3-1-2022

Reversible Data Hiding with a New Local Contrast Enhancement Approach

Eduardo Fragoso-Navarro
Universidad Nacional Autónoma de México

Manuel Cedillo-Hernandez
Instituto Politécnico Nacional

Francisco Garcia-Ugalde
Universidad Nacional Autónoma de México

Robert Morelos-Zaragoza
San Jose State University, robert.morelos-zaragoza@sjsu.edu

Recommended Citation

Eduardo Fragoso-Navarro, Manuel Cedillo-Hernandez, Francisco Garcia-Ugalde, and Robert Morelos-Zaragoza. "Reversible Data Hiding with a New Local Contrast Enhancement Approach" *Mathematics* (2022). https://doi.org/10.3390/math10050841

mathematics

MDPI

Article

Reversible Data Hiding with a New Local Contrast Enhancement Approach

Eduardo Fragoso-Navarro [1], Manuel Cedillo-Hernandez [2,*], Francisco Garcia-Ugalde [1] and Robert Morelos-Zaragoza [3]

[1] Facultad de Ingenieria, Universidad Nacional Autonoma de Mexico (UNAM), Av. Universidad No. 3000, Ciudad Universitaria, Coyoacan, Mexico City 04510, Mexico; efragoson1400@alumno.ipn.mx (E.F.-N.); fgarciau@unam.mx (F.G.-U.)
[2] Instituto Politecnico Nacional (IPN), Escuela Superior de Ingenieria Mecanica y Electrica, Unidad Culhuacan, Av. Santa Ana No. 1000, San Francisco Culhuacan, Coyoacan, Mexico City 04430, Mexico
[3] College of Engineering, San Jose State University, San Jose, CA 95192, USA; robert.morelos-zaragoza@sjsu.edu
* Correspondence: mcedilloh@ipn.mx

Abstract: Reversible data hiding schemes hide information into a digital image and simultaneously increase its contrast. The improvements of the different approaches aim to increase the capacity, contrast, and quality of the image. However, recent proposals contrast the image globally and lose local details since they use two common methodologies that may not contribute to obtaining better results. Firstly, to generate vacancies for hiding information, most schemes start with a preprocessing applied to the histogram that may introduce visual distortions and set the maximum hiding rate in advance. Secondly, just a few hiding ranges are selected in the histogram, which means that just limited contrast and capacity may be achieved. To solve these problems, in this paper, a novel approach without preprocessing performs an automatic selection of multiple hiding ranges into the histograms. The selection stage is based on an optimization process, and the iterative-based algorithm increases capacity at embedding execution. Results show that quality and capacity values overcome previous approaches. Additionally, visual results show how greyscale values are better differentiated in the image, revealing details globally and locally.

Keywords: contrast enhancement; histogram shifting; information security; image processing

check for updates

Citation: Fragoso-Navarro, E.; Cedillo-Hernandez, M.; Garcia-Ugalde, F.; Morelos-Zaragoza, R. Reversible Data Hiding with a New Local Contrast Enhancement Approach. *Mathematics* **2022**, *10*, 841. https://doi.org/10.3390/math10050841

Academic Editors: Akemi Galvez Tomida, Lihua You, Hassan Ugail, Andres Iglesias Prieto and Alexander Malyshev

Received: 3 February 2022
Accepted: 5 March 2022
Published: 7 March 2022

1. Introduction

Reversible data hiding (RDH) refers to concealing information in a host image and subsequently recovering both original elements. Reversibility is necessary, e.g., in medical and military areas or in any field in which permanent distortions of the image are not permitted. We can identify three basic types of RDH approaches [1], including lossless compression (LC) [2–4], difference expansion (DE) [5–8], and histogram shifting (HS) [9–15]. Previous methods have been combined to substantially improve results [8,14,16–20]. In the context of HS schemes, the first HS technique described in [9] selects a range in the histogram between a peak-bin and a zero bin, and then, the bins in the open interval are shifted toward the zero bin to create a vacancy. Finally, the binary information is hidden between the peak bin and the adjacent empty bin. The hiding rate in bits is equal to the number of pixels with the maximum occurrence, which corresponds to the peak bin. Additionally, since the values of the pixel are shifted just by one, the hidden information is imperceptible.

The development of new RDH techniques mainly seeks an increment of the hiding rate with the preservation of the perceptual quality of the image. Nevertheless, some distortion might appear if a higher amount of information is embedded. To the best of our knowledge, authors in [21] discovered that HS techniques could increase contrast as a favorable distortion when embedding higher payloads and propose a method called RDH with contrast enhancement (RDH-CE). This approach performs the shifting of the histogram bins multiple times in both directions, increasing the hiding rate and contrast of the image. The result of [21] is similar to that obtained in histogram equalization (HE) [22,23] methods because the peak values of the histogram are split and reduced, and the bins are more distributed. To avoid overflows caused by shifting, a preprocessing method merges bins and generates vacancies on both sides of the histogram. Two drawbacks in RDH-CE schemes are generated by the preprocessing:

- Firstly, since the merged bins are not adjacent, the change in pixel values may cause some visual artifacts in the image;
- Secondly, the position of the merged pixels must be saved to ensure reversibility; thus, a location map is generated and hidden into the histogram. Since the location map is the size of the image, it must be compressed; nevertheless, the payload capacity is reduced by the additional hidden information.

Based on the algorithm in [21], in the scientific literature, some RDH-CE techniques had been developed to increase capacity, improve the enhancement of contrast, and reduce undesirable distortions [24–31]. Most of the RDH-CE [21,26–31] methods are focused on alleviating preprocessing drawbacks, such as distortion and capacity reduction caused by the location map. From our perspective, we consider that RDH-CE approaches without preprocessing are not explored sufficiently; moreover, the scheme in [25] has some important mechanisms that can be improved and extended. The proposed RDH-CE scheme of [25] selects a peak bin and a minimum bin in the histogram, avoiding the constraint of selecting a zero bin used in the HS approach. The range selection and embedding processes are iterated until all information is hidden. The shifting of the histogram at each iteration overlaps the minimum bin with its adjacent bin; therefore, it is necessary to save the location of the pixels whose values were overlapped. This information is contained in a location map that is hidden in conjunction with the payload. Since location maps are obtained from minimum bins, they are expected to be small.

In general, RDH schemes aim to simultaneously increase the capacity, contrast, and quality of the image. The present proposal was designed to improve these characteristics. On the one hand, previous RDH schemes [21,24–31] hide the information in a few ranges of the histogram image. In contrast, we propose an algorithm that selects the greatest number of ranges possible in each histogram. The capacity can be increased since more ranges can hide more information. On the other hand, contrast is not just incremented but also visually improved since more ranges contain more peak bins that are spread and can be differentiated in the pixels of the contrasted image. Additionally, more ranges reduce the number of intermediate bins that are just shifted and decrease quality with no contribution to contrast. Furthermore, our proposal applies to optimal range selection on the raw histogram, avoiding preprocessing with unnecessary quality drops. Finally, the capacity is not restricted in advance because the algorithm is iterated over previous contrasted images, increasing the capacity in embedding execution. In this way, we denominate the scheme as multirange-level histogram shifting (MRLHS), considering that each iteration is called level, and the embedding process is achieved in multiple ranges of the histogram.

To adapt the proposal for practical use, we collect all the removal information, such as the peak bins, minimum bins, and other side data in a binary string. This information is hidden into the improved image to increase the contrast further and generate a smaller binary sequence to recover the information.

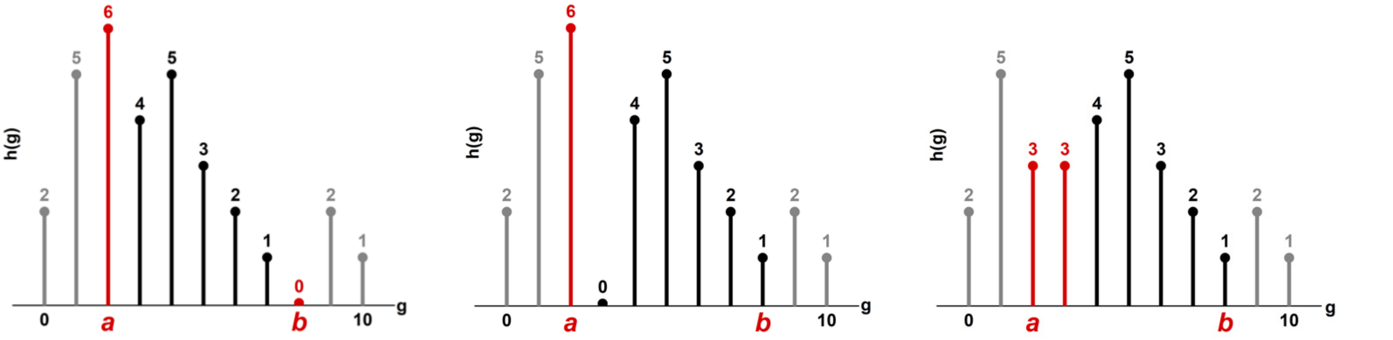
Results show that SSIM and PSNR qualities are better than previous approaches for almost all test images. Additionally, the capacity is considerably improved under similar contrast levels. Visual results show that the enhanced image better represents the local content of the original image, disclosing objects in the darkest and brightest areas and improving texture appearance.

The rest of this paper is organized as follows. Section 2 describes the basic HS method with zero bin. The proposed scheme is described in Section 3 and includes the HS with minimum bin (HS-MB) approach, the selection of the optimum set of ranges, and the generation of the removal sequence. The experimental results are shown in Section 4, discussion in Section 5, and conclusions are presented in Section 6.

2. Background

2.1. Embedding Process of the Conventional Histogram Shifting

Histogram shifting [9] selects a peak bin a and a zero bin b of the image histogram, as shown in Figure 1a, and embeds the binary information. In case $a > b$, the pixels of the image with values that are in the range $[a+1, b]$ are increased by 1; consequently, the histogram bins are shifted, as shown in Figure 1b. The peak bin a and the new vacant bin $a+1$ are selected to embed a binary string. A pixel of the image with a value is increased by 1 or remains intact if the bit of the string that we want to embed is 1 or 0, respectively. If we have a binary string with six elements and the same number of $0'$s and $1'$s, bins a and $a+1$ will have the same histogram value as shown in Figure 1c. Additionally, the capacity of the range selected in Figure 1 is 6 bits, which is equivalent to the histogram frequency $h(a)$ in the peak bin a.



(**a**) (**b**) (**c**)

Figure 1. Histogram embedding in histogram shifting scheme; (**a**) original histogram; (**b**) shifted histogram; (**c**) histogram with concealed data.

Using (1), the hiding process is applied in each pixel to hide a bit of information. The embedding in the pixels can be in a raster scan order, but not necessarily.

$$J(m,n) = \begin{cases} I(m,n) + d, & \min(a,b) < I(m,n) < \max(a,b) \\ I(m,n) + d[B(q)], & I(m,n) = a \\ I(m,n), & I(m,n) < \min(a,b) \textit{ or } I(m,n) > \max(a,b) \end{cases}, \tag{1}$$

From (1), $I(m,n)$ and $J(m,n)$ are the original and the marked pixels, respectively, in the (m,n) coordinates of the image. $B(q)$ is the q-th consecutive bit of the binary string B that we want to hide in the pixel. To generalize, the coefficient d determines the shifting direction, which must be from a to b, according to (2).

$$d = \begin{cases} +1, & a < b \\ -1, & a > b \end{cases}, \tag{2}$$

2.2. Reversibility Process of the Conventional Histogram Shifting

To read the binary string concealed in the image, we need to know the peak bin value a and the peak bin value b. Then, we read the hidden binary string B' from the image by applying (3) to each pixel in the coordinates (m,n). The reading order of the pixels must be in the order used in the embedding.

$$B'(q) = \begin{cases} 0, & J(m,n) = a \\ 1, & J(m,n) = a + d \end{cases}, \tag{3}$$

Finally, to obtain the recovered image I' from the marked image J we read each pixel in the position (m,n) in the correct order and then apply (4).

$$I'(m,n) = \begin{cases} J(m,n) - d, \ \min(a,b) < I(m,n) < \max(a,b) \\ J(m,n) - d\left[B'(q)\right], \ I(m,n) = a \\ J(m,n), \ I(m,n) < \min(a,b) \ or \ I(m,n) > \max(a,b) \end{cases}, \quad (4)$$

2.3. Reversible Data Hiding with Contrast Enhancement

The scheme in [25] can help to understand the basis of RDH-CE. Figure 2a shows a representation of an original histogram corresponding to an original image (Figure 2d). The preprocessing stage consists of selecting, shifting, and overlapping the several outer bins with the neighbor bins, as shown in Figure 2b. Then, two peak bins are identified and related with two corresponding zero bins that were generated in the preprocessing. In the embedding stage, histogram shifting is applied to each peak bin and zero bin pair. The process is repeated until there are no more vacancies in the outer bins. The resulting histogram is flatter (Figure 2c), and the contrast of the corresponding image is increased (Figure 2e), similar to histogram equalization approaches [22,23].

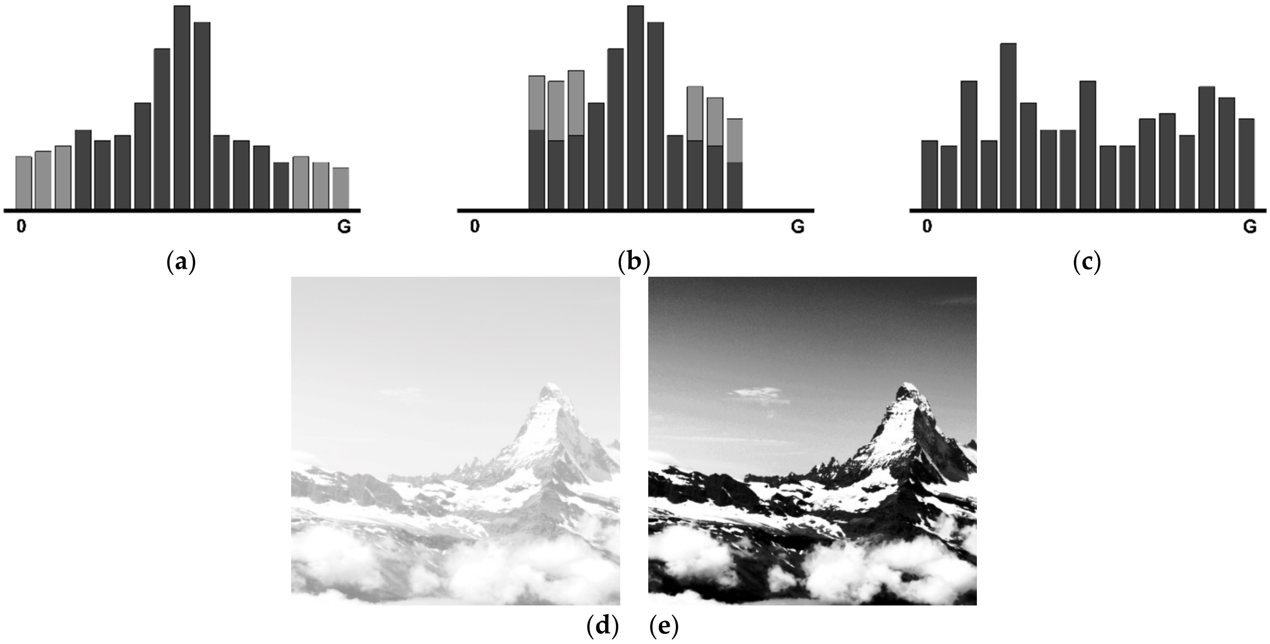


Figure 2. Representation of the (**a**) original, (**b**) preprocessed, (**c**) final histograms. (**d**) Original, and (**e**) contrasted images [25].

Based on the general RDH description, schemes in [21,24–31] propose different preprocessing, peak bin selections, or histogram generation stages in order to obtain better results.

3. Proposed RDH-CE Based on MRLHS

As shown in Figure 3, the embedding stage of the proposed algorithm consists of two MRLHS embedding blocks, each one followed by a removal sequence generator. Subsequently, the removal stage reverses the embedding process by applying two MRLHS removal blocks preceded by key readers.

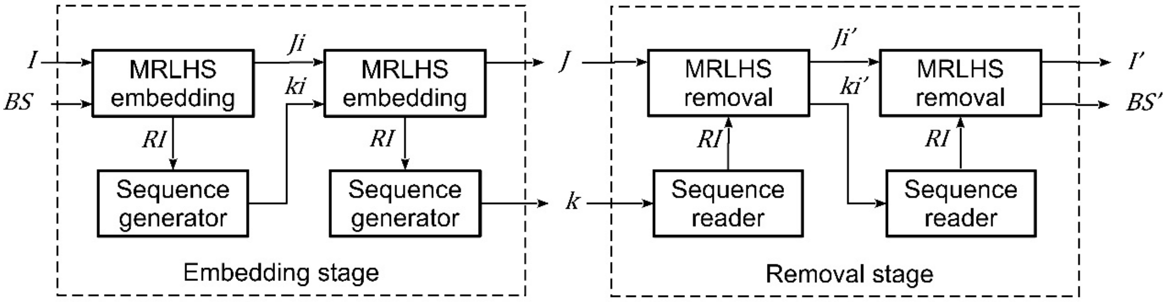


Figure 3. Embedding and removal stages based on MRLHS.

More specifically, the embedding stage is composed of the following four steps:

1. Embed the binary string BS into the image I using the MRLHS embedding process. This produces an enhanced internal image Ji and the removal information RI;
2. Generate internal removal sequence ki from the removal information RI;
3. Embed internal removal sequence ki into the image Ji. This produces the enhanced image J and the new removal information RI;
4. Generate external removal sequence k with the new removal information RI.

The data in removal information RI are binarized and placed into a removal sequence by a sequence generator, and the sequence reader recovers the information, as shown in Figure 3. The MRLHS embedding process and the sequence generator block are repeated twice for two reasons. The first one is that embedding the internal sequence ki in Ji further increases the contrast of the image so that contrast of image J is greater than the contrast of image Ji. Secondly, internal removal sequence ki may be too large for practical use; by repeating the process, we obtain a smaller sequence k which is delivered to the user. In this way, ki contains the removal information generated by embedding the binary string BS in I, and k contains the removal information generated by embedding ki in Ji. The authorized user must have just the smaller sequence k to reverse all the embedding processes.

The sequences k and ki are just a binary representation of the concatenated removal information. We claim that proposing how to construct both sequences and know the length of the binary representation after the embedding process have some advantages. Firstly, the user can have an easier way to construct sequence ki, concatenate it to the payload BS, and embed them into the image. Secondly, we can easily know the size of sequence k so the user can decide how to manage this information. Finally, some algorithms can be applied to the binary sequences, such as encryption, to increase security. However, as most of the RDH approaches [21,24–31], the usage of security techniques is not included in the proposals since it depends on the user preferences and applications. One characteristic that may add some level of security in the present algorithm is that sequences k and ki dynamically change according to the image, the information, and other parameters that are not known in advance.

On the other hand, the removal stage reverses the process of the embedding stage as shown in Figure 3 and is composed of the following four steps:

1. Read the information of the external removal sequence k to obtain removal information RI;
2. Extract the internal removal sequence ki and obtain an internally enhanced image Ji';
3. Extract the information of the internal removal sequence ki to obtain new removal information RI;
4. Extract the binary vector BS' and the image I', which experimental results demonstrate are equal to the binary string BS and image I, respectively.

The general MRLHS embedding and removal blocks of Figure 3 are detailed in Figure 4a,b, respectively. The authors in [15] proved that using HS to embed information into blocks rather than into the complete image increases the ratio between visual quality and capacity. Therefore, we first divide the image I into $F \times C$ nonoverlapped blocks and obtain the histogram of each block, as shown in Figure 4a. To increase capacity, we select multiple ranges of each histogram and embed a segment of information at each range. Finally, to increase capacity in embedding execution, we repeat the process many times at subsequent iterations, which we call levels. Each level L conceals part of the information BS and further increases the contrast of the image because of the histogram redistributions. The algorithm repeats the embedding level until BS is totally hidden and finally obtains an enhanced image, as well as the removal information RI. The data in RI contain the parameters F, C, L, the length of BS, and the selected ranges, which are used for the reversibility of the proposed method, as illustrated in Figure 4b.

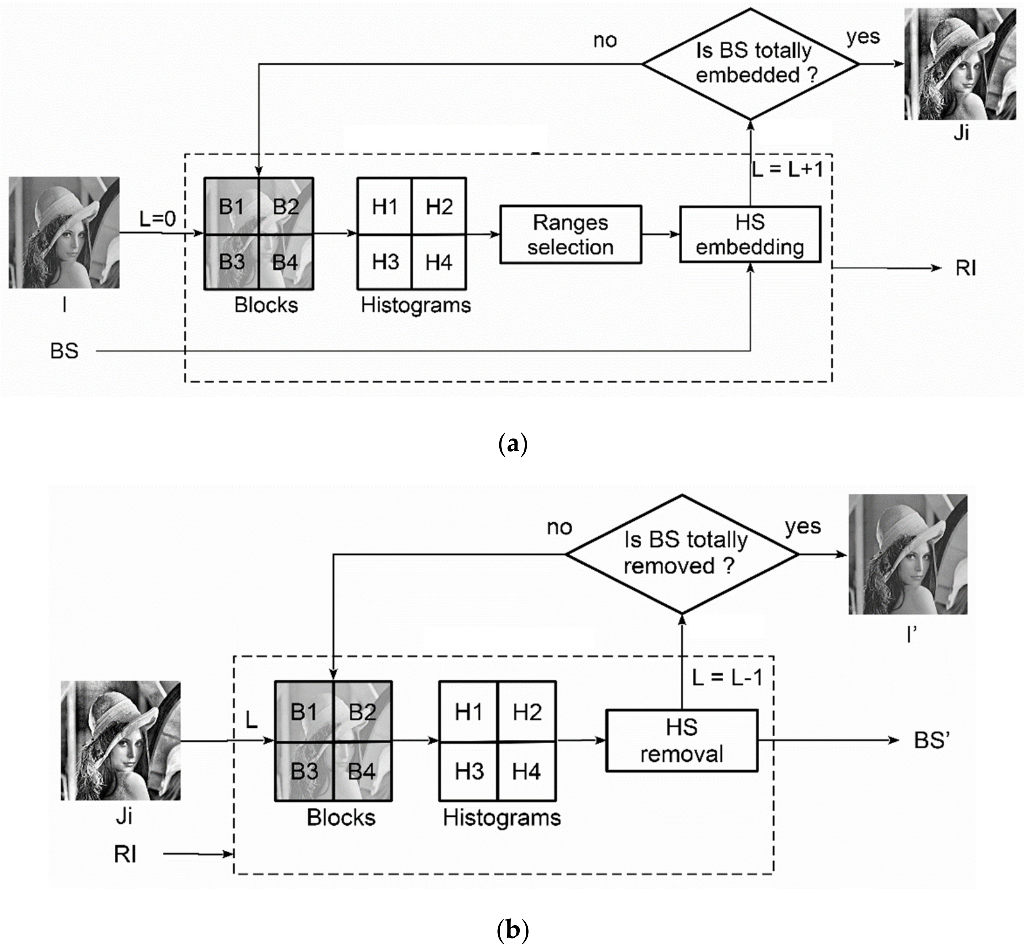
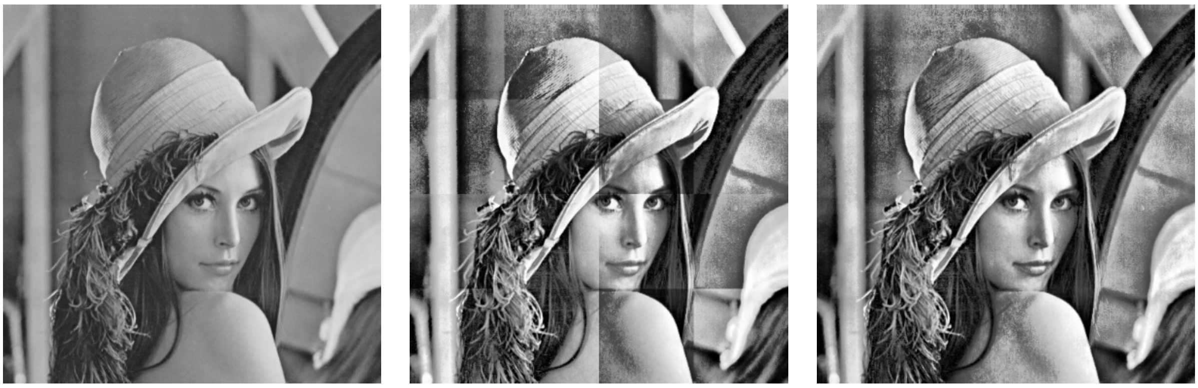


Figure 4. Detailed MRLHS process: (**a**) embedding and (**b**) removal.

In the proposed RDH embedding stage of Figure 4a, the contrast of the image is enhanced, but an undesirable block effect may appear and grow when the embedding level increases, as shown in Figure 5b. This effect occurs if the blocks are selected at the same position in the progressive levels. Then, to diminish this effect and at the same time improve the visual quality of the enhanced image, we propose shifting the position of the blocks at each increment of level L. First, starting with the original image, we select the blocks considering null shifting and proceed to the first data hiding round, obtaining an enhanced image of level $L = 1$. Then, if BS data are not totally embedded into the enhanced image, we select the blocks shifted vertically and horizontally the amount of 1/2 their vertical and horizontal size, respectively; if another round of data hiding is needed, then the next shifting step considers a ratio of 1/4 and so on. Therefore, the ratios to shift the block positions in each level are based on the sequence (0, 1/2, 1/4, 3/4, 1/8, 3/8, 5/8, 7/8, 1/16, 3/16, 5/16, 7/16 . . . .). The shifting sequence avoids the overlapping of the block edges at each embedding level and reduces the block effect. For illustrative purposes, in

Figure 5c, we show several advantages when performing block shifting selection. The total levels needed for embedding are less, and therefore, quality increases and processing time decreases.



(**a**)

(**b**)
Hiding rate = 1 bpp
Total levels = 62
PSNR = 17.3078
SSIM = 0.6858
Embedding time = 199 s

(**c**)
Hiding rate = 1 bpp
Total levels = 49
PSNR = 19.7412
SSIM = 0.7078
Embedding time = 165 s

Figure 5. (**a**) Original image, (**b**) enhanced image without block shifting, and (**c**) enhanced image with block shifting.

Algorithms 1 and 2 show pseudocode that can help to implement the present proposal.

Algorithm 1: Embedding stage (Figure 3)

```
Inputs: Image $I$ and binary string $BS$
Outputs: Enhanced image $J$ and removal sequence $k$
MRLHS embedding for BS (Figure 4a):
    Initialize enhanced image $Ji = I$ and level $L = 0$
    while $BS$ is not totally embedded
        Upgrade $L = L + 1$
        Divide image $Ji$ into $F \times C$ blocks (with block shifting)
        Calculate histogram of each block
        Select ranges of each histogram (Section 3.2)
        Embed a segment of $BS$ into each range of $Ji$ (Section 3.1)
        Identify data for removal information $RI$: (a) $F, C$ (b) peak bins and minimum bins
        (c) $L$ (d) $BS$ length
    end while
    Generate internal removal sequence ki from RI (Section 3.3.1)
MRLHS embedding for $ki$ (Figure 4a):
    Initialize enhanced image $Ji = I$ and level $L = 0$
    while $ki$ is not totally embedded
        Upgrade $L = L + 1$
        Divide image $Ji$ into $F \times C$ blocks (with block shifting)
        Calculate histogram of each block
        Select ranges of each histogram (Section 3.2)
        Embed a segment of $ki$ into each range of $J$ (Section 3.1)
        Identify data for removal information $RI$: (a) $F, C$ (b) peak bins and minimum bins
        (c) $L$ (d) $BS$ length
    end while
    Generate internal removal sequence $k$ from $RI$ (Section 3.3.1)
```

Algorithm 2: Removal stage (Figure 3)

```
Inputs: Enhanced image $J$ and removal sequence $k$
Outputs: Recovered image $I'$ and recovered binary string $BS$
Read the removal information $RI$ from removal sequence $k$ (Section 3.3.2)
MRLHS removal for $ki$ (Figure 4b):
    Initialize recovered image $Ji' = J$
    while $BS$ is not totally removed ($L \neq 0$)
        Divide image $Ji'$ into $F \times C$ blocks (with block shifting)
        Calculate histogram of each block
        Remove a segment of $ki'$ from each range defined with corresponding peak bin $a$
        and min-bin $b$ (Section 3.1.2)
        Upgrade $L = L - 1$
    end while
    Read the removal information $RI$ from removal sequence $ki'$ (Section 3.3.2)
MRLHS removal for $BS$ (Figure 4b):
    Initialize recovered image $I' = Ji'$
    while $ki$ is not totally removed ($L \neq 0$)
        Divide image $I'$ into $F \times C$ blocks (with block shifting)
        Calculate histogram of each block
        Remove a segment of $BS$ from each range defined with corresponding peak bin $a$
        and min bin $b$ (Section 3.1.2)
        Upgrade $L = L - 1$
    end while
    Concatenate segments to recover $BS$ (Section 3.3.1)
```

In Section 3.1, we explain the embedding process at each range, which is an improvement of the proposal in Section 2. The proposed HS approach considers a range with a minimum bin instead of a zero bin. This allows us to increase the number of ranges into one histogram and, therefore, obtain more capacity and contrast. In Section 3.2, we explain the selection of the optimum set of ranges with minimum bins to hide the information in the histograms. This process is achieved using a decision table that optimizes the trade-off between capacity and quality. Finally, in Section 3.3, we propose the removal sequence generator and reader blocks shown in Figure 3.

3.1. Embedding Process of the Conventional Histogram Shifting

If we divide the histogram into ranges, we may find just a few with zero bin as described in Section 2.2. Therefore, the proposal in [25] selects a range that has minimum bins with values different from zero. In this way, we can select more than one range and increase the capacity of the histogram.

However, in the shifting process, the minimum bin and its adjacent bin are overlapped. Therefore, to achieve the complete reversibility of the hiding process, it is necessary to save the positions of the pixels in which values are merged. In our HS-MB, we propose to generate a header with the position information and hide it with the payload into the corresponding range.

3.1.1. HS-MB Embedding

If the histogram does not have a zero bin b to define a range, we propose selecting a minimum bin b whose histogram frequency $h(b)$ is equal or different from zero, as shown in Figure 6a, where $h(b) = 1$. Then, the range $[a+1, b-1]$ is shifted, as shown in Figure 6b, and the binary string is concealed in the bins a and $a+1$ according to (1). As an example, if we hide a binary string with two zeros and four ones, the histogram changes, as shown in Figure 6c.

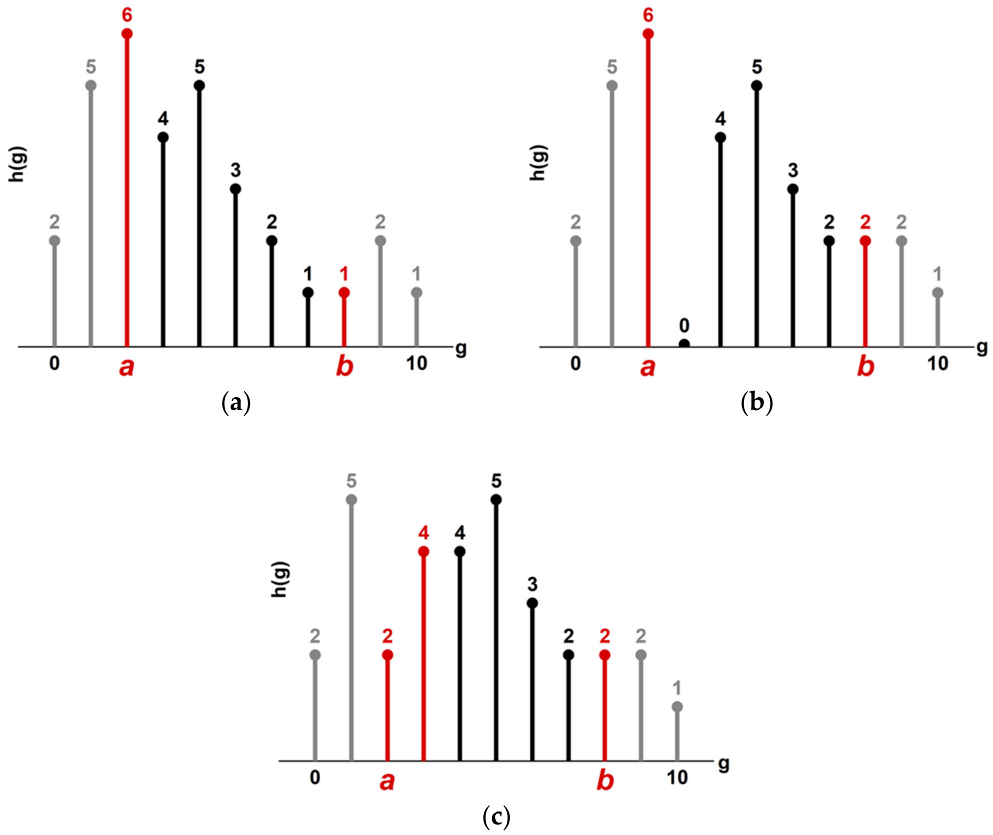


Figure 6. Proposed embedding procedure with a minimum bin: (**a**) original histogram; (**b**) shifted histogram; (**c**) histogram with concealed data.

When the range $[a+1, b-1]$ in Figure 6a is shifted, the bins $b-1$ and b overlap as shown in Figure 6b, where $h(b) = 2$. This means that the pixels in the original image with values $b-1$ and b equal to b after the shifting process. To completely recover the image in the removal process, it is necessary to know which pixels were equal to $b-1$ and b before shifting. Therefore, we propose to save this information in a header H and concatenate it before the payload B. Then, the concatenated string HB is concealed in the bins a and $a+1$.

Therefore, we need to define the elements of H that indicate the original pixel values of the overlapped bins. To generalize, value d in (2) indicates the shifting direction and defines de minimum bin b and its adjacent bin $b-d$. The first two bits of H will indicate the original state of bins $b-d$ and b. If the bin in $b-d$ is a zero bin, then the first bit $H(1)$ equals 0, and 1 otherwise, as shown in (5). If the bin in b is a zero bin, then the second bit $H(2)$ equals 0, and 1 otherwise, as shown in (6). The next bits in H will indicate which pixel values of the original image were $b-d$ or b before shifting. The pixels of the original image are scanned in a defined order, and when the p-th consecutive pixel with value $b-d$ or b is found, the next bit $H(2+p)$ is set to 1 or 0, respectively, as shown in (7).

$$H(1) = \begin{cases} 0, & h(b) = 0 \\ 1, & h(b) \neq 0 \end{cases}, \tag{5}$$

$$H(2) = \begin{cases} 0, & h(b-d) = 0 \\ 1, & h(b-d) \neq 0 \end{cases} \tag{6}$$

$$H(2+p) = \begin{cases} 0, & I(m,n) = b \\ 1, & I(m,n) = b-d \end{cases} \tag{7}$$

The binary string HB that concatenates the header H and the information B is defined in (8). If the first two elements of H are equal to 1, it means that there are overlapped bins after shifting, and we must save the rest of the string H; otherwise, it is not necessary. This strategy considerably reduces the length of the header H in case we do not have overlapped pixels to be differentiated.

$$HB = \begin{cases} [H,B], \ H(1) = 1 \, y \, H(2) = 1 \\ [H(1), H(2), B], \ \text{otherwise} \end{cases}, \tag{8}$$

In the example of Figure 6a, $H(1) = 1$ and $H(2) = 1$, the rest of H has a length of 2 because we have two overlapped pixels that we must distinguish in the removal process. These two values $H(3)$ and $H(4)$ depend on which pixel is found first in the original image, $b - d$ or b, in the scanning order. For example, if we find b value first, then $H(3) = 1$ and $H(4) = 0$. Therefore, the total length of H is two plus the number of pixels with overlapped values, which is $h(b) + h(b-d) + 2$.

Since we must conceal the information contained in H, the net capacity of the range is reduced. For the example in Figure 6, the absolute capacity is $h(a) = 6$ but 4 bits are used to conceal the header H, and consequently, we have a net capacity equal to 2 bits. Hence, the net capacity C is the absolute capacity $h(a)$ of the range minus the length of H as follows:

$$C = \begin{cases} h(a) - [h(b) + h(b-d) + 2], \ H(1) = 1 \, \mathit{and} \, H(2) = 1 \\ h(a) - 2, \ \text{otherwise} \end{cases}, \tag{9}$$

Once we have HB, the embedding process is achieved using (1) with HB instead of B, and the length of B must be less or equal to the net capacity C.

Finally, we define the distortion D of the range as the number of pixels affected by the shifting process. This value is defined in (10) as the sum of all the histogram values in the range $[\min(a,b) + 1, \max(a,b) - 1]$.

$$D = \sum_{g=\min(a,b)+1}^{\max(a,b)-1} h(g), \tag{10}$$

3.1.2. HS-MB Removal

The removal process is divided into two steps, the reading of the concealed information and the image recovering. For the first step, each consecutive q-th value of the extracted string HB' is obtained by reading the value of each pixel in the position (m,n) in the order used in the embedding process, as follows:

$$HB'(q) = \begin{cases} 0, \ J(m,n) = a \\ 1, \ J(m,n) = a + d \end{cases}, \tag{11}$$

The recover payload B' is obtained by reading the elements of HB' from the position after the header, as follows:

$$B' = \begin{cases} HB'(3 + h(b), \ldots, length(HB')), \ HB'(1) = 1 \, \mathit{and} \, HB'(2) = 1 \\ HB'(3, \ldots, length(HB')), \quad \text{otherwise} \end{cases}, \tag{12}$$

To obtain the recovered image I' we must read each pixel $J(m,n)$ of the marked image J in the order defined in the embedding process, and calculate each pixel $I'(m,n)$ as follows:

$$I'(m,n) = \begin{cases} J(m,n) - d, \ \mathit{Condition} \ 1 \\ J(m,n) - d\left[HB'(p+2)\right], \ \mathit{Condition} \ 2 \\ J(m,n), \ \mathit{Condition} \ 3 \end{cases}, \tag{13}$$

where p is the consecutive number of the found pixel whose value equals b.

The conditions given in (10) are defined as:

$\mathit{Condition} \ 1:$

$[\min(a,b) < J(m,n) < \max(a,b)] \ \mathit{or} \ [J(m,n) = b \ \mathit{and} \ HB'(1) = 0 \ y \ HB'(2) = 1]$

$Condition\ 2:$
$J(m,n) = b\ and\ HB'(1) = 1\ and\ HB'(2) = 1$

$Condition\ 3:$
$[J(m,n) < \min(a,b)\ or\ J(m,n) > \max(a,b)]\ or\ [J(m,n) = b\ and\ HB'(1) = 1\ and\ HB'(2) = 0]$

The first condition corresponds to the bins that are shifted, as the black ones in Figure 6a. The second condition separates bins in case they are overlapped. Finally, condition 3 corresponds to the bins that must remain, as represented in gray lines in Figure 6.

3.2. Selection of Optimal Set of Ranges

In this section, we propose to select a set of ranges per histogram and conceal information in each one of them using the HS-MB technique. This may have two advantages: first, we can increase the total capacity of hidden information, and secondly, the marked histogram will be more distributed, which will cause a better increment in the image contrast.

Better contrast increment is one of the main advantages when selecting multiple ranges at one embedding iteration. Proposals in [21,24–31] select one or few ranges, and the contrast is only incremented by the few peak bins that spread to their adjacent bins. Additionally, since the difference between the peak bin and the minimum bin tend to be large, all the intermediate bins that are shifted do not contribute to increasing the contrast ratio. Therefore, selecting more ranges increases the number of peak bins that contribute to improving contrast and decreases the number of intermediate bins. As a result, more grayscale values can be differentiated from each other in the image after the embedding process. In Section 4.5, we compare the results of both approaches and show that selecting many ranges contributes to increasing the local details of the image.

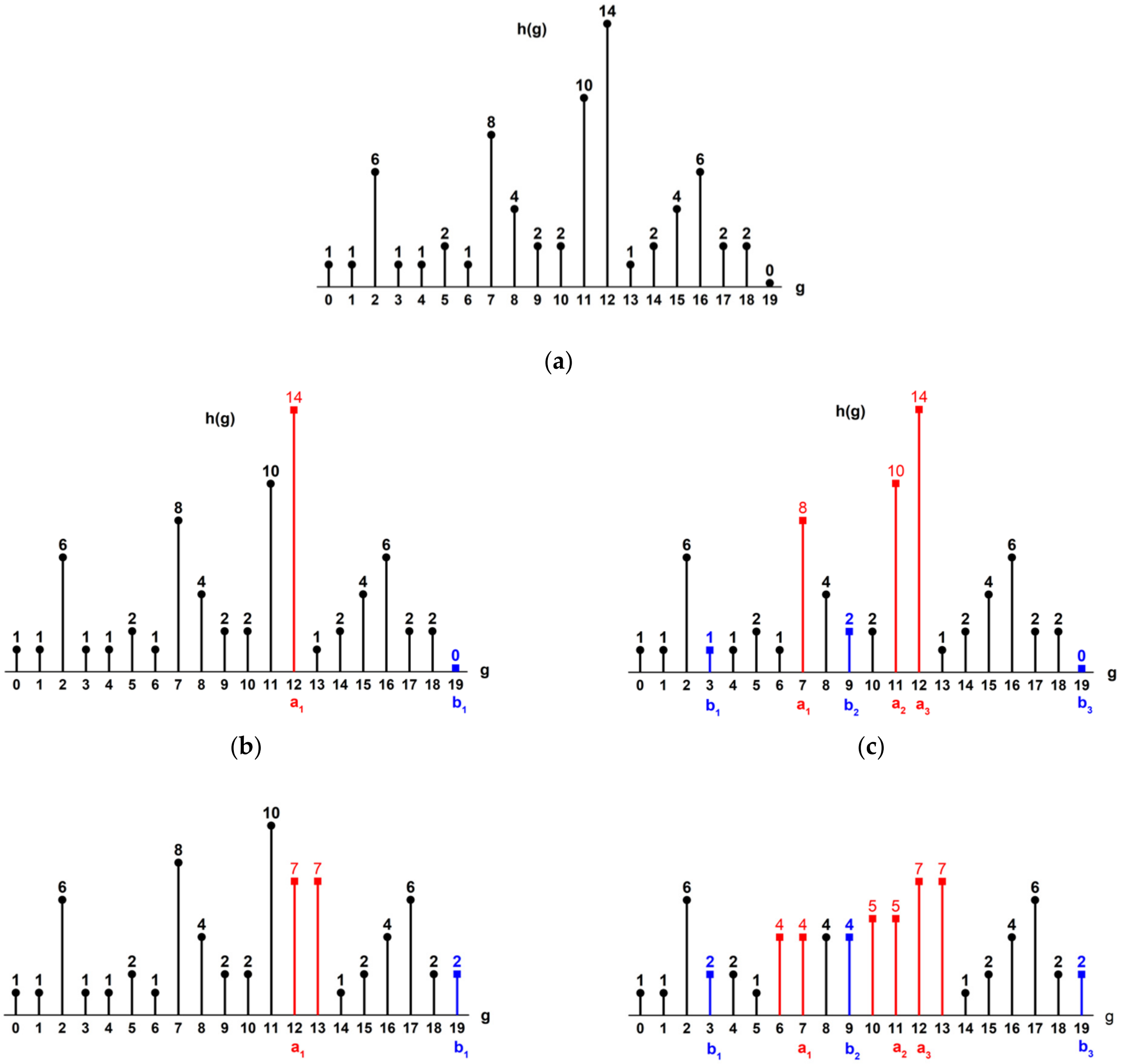
We use the histogram in Figure 7a as an example. In Figure 7b, we selected one range with the peak bin a_1 and the zero bin b_1 that define the range whose capacity is equivalent to 14 bits using the HS zero bin technique. On the other hand, in Figure 7c, we select three peak bins a_1, a_2, and a_3 with their corresponding minimum bins b_1, b_2, and b_3, respectively. The capacity of each range according to (9) is 4, 4, and 12 bits, respectively, and the total capacity is 20 bits, greater than the capacity of the range selected in Figure 7b. To show the histogram distribution, Figure 7d,e shows the marked histograms using the ranges of Figure 7b,c, respectively. We can see that the histogram in Figure 7e is less sharp than that of Figure 7d since more peak bins were distributed.

Since there is more than one range, we define the total capacity C_T of the set with p ranges as the sum of the individual capacities of all ranges as follows:

$$C_T = \sum\nolimits_{s=1}^{p} C_s, \tag{14}$$

where C_s is the capacity of the s-th range calculated by (9).

The number of sets of ranges that we can use to hide information may be large. For example, if we select in the histogram of Figure 7a the peak bin a with value 12, we can select a minimum bin b equal to 0, 1, 2 3, 4, 5, 6, 8, 9, 14, 15, 16, 17, 18, or 19, so we have 15 possible options. Using an exhaustive calculation for the histogram of Figure 7a, we found that the number of possible range sets using just one peak bin at a time (a equals 2, 7, 11, 12, or 16) is 59. If we add the number of possible sets using two to five peak bins, we obtain 1509 possibilities. One of the main objectives of the present proposal is to choose the better possibility, the one that has the maximum capacity with the minimum distortion to the image. Therefore, finding all possibilities and selecting the best may be exhaustive for histograms with bins in the range 0–255. To solve this problem, we propose an optimization method that selects an initial range and updates it with a new set of ranges that increase capacity at each iteration.



(**a**)

(**b**) (**c**)

(**d**) (**e**)

Figure 7. (**a**) Example of a histogram; (**b**) one range selected with capacity equal to 14 bits; (**c**) three ranges selected with total capacity equal to 20 bits; (**d**) Histogram with concealed information in the range of (**b**); (**e**) Histogram with concealed information in the three ranges of (**c**).

Optimization Process

In this section, we explain the main iterative process to find the best set of ranges. We will define two vectors $\boldsymbol{a} = [a_1, a_2, \ldots a_s, \ldots, a_p]$ and $\boldsymbol{b} = [b_1, b_2, \ldots b_s \ldots, b_p]$ whose p elements are the peak bins and minimum bins, respectively, arranged in descending order. For example, the vectors for the range in Figure 7b are $\boldsymbol{a} = [a_1] = [12]$ and $\boldsymbol{b} = [b_1] = [19]$; on the other hand, the vectors of the ranges in Figure 7e are $\boldsymbol{a} = [a_1, a_2, a_3] = [7, 11, 12]$ and $\boldsymbol{b} = [b_1, b_2, b_3] = [3, 9, 19]$.

The iterative process is shown in Figure 8 and described as follows. First, we initialize vector $\boldsymbol{a}$ with the bin of the maximum value $\max\{h(\boldsymbol{g})\}$ of the histogram, and vector $\boldsymbol{b}$ with the bin that corresponds to the minimum value $\min\{h(\boldsymbol{g})\}$. Having the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, we select the next peak bin a_t different from the previously selected. Using the decision Table 1, we find the minimum bin b_t and incorporate a_t and b_t in new vectors $\boldsymbol{a_t}$ and $\boldsymbol{b_t}$. If the total capacity C_T that corresponds to vectors $\boldsymbol{a_t}$ and $\boldsymbol{b_t}$ is greater than the total capacity C_T of the current vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, we update them as follows:

$$\begin{aligned} \boldsymbol{a} &= \boldsymbol{a_t}, \\ \boldsymbol{b} &= \boldsymbol{b_t}, \end{aligned} \tag{15}$$

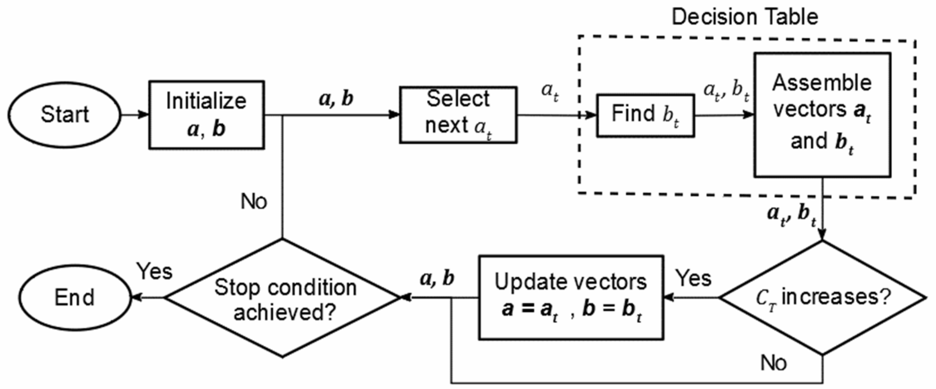

Decision Table
Start
Initialize a, b
a, b
Select next a_t
a_t
Find b_t
a_t, b_t
Assemble vectors $\boldsymbol{a}_t$ and $\boldsymbol{b}_t$
$\boldsymbol{a}_t$, $\boldsymbol{b}_t$
No
End
Yes
Stop condition achieved?
a, b
Update vectors $\boldsymbol{a} = \boldsymbol{a}_t$, $\boldsymbol{b} = \boldsymbol{b}_t$
Yes
C_T increases?
No

Figure 8. Optimization process based on iterative updating of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$.

If the capacity were not improved, we would select the next peak bin a_t and repeat the process. The process continues until one of the following conditions occurs: the desired capacity is reached, the capacity does not increase anymore, or the stop condition $h(a_t) \leq Vp\max\{h(g)\}$ is achieved. The stop value $Vp \in [0,1]$ may stop iterations; if we decrease the value, the range selection time decreases, but the total capacity may not be the maximum possible.

The decision table finds the new minimum bin b_t and assign it to a_t or to a previous peak bin, according to some conditions. First, we find in the table which condition a_t complies in conjunction and then calculate the corresponding test minimum bin b_t and the test vectors $\boldsymbol{a_t}$ and $\boldsymbol{b_t}$. The first condition refers to the position of a_t in the histogram with respect to other bins in vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. The second condition assures that the peak bin a_t is sufficiently separated from another peak bin to search between them for the new minimum bin b_t. The third condition looks for previously selected bins close to a_t to decide if b_t will be assigned to a_t or to another peak bin. There are some important considerations. Firstly, some cases just need the compliance of one or two conditions. Secondly, if a_t complies with more than one set of conditions, we must choose the one that improves capacity C_T.

Additionally, we will define a programming function that searches the best minimum bin b_t for a given peak bin a in a defined range $[liminf, limsup]$ as follows:

$$b_t = minb(liminf, limsup, a), \tag{16}$$

The Equation (16) must discard the bins in the range $[a-d, a+d]$ since we need at least two bins (a and adjacent) to conceal the information. If the function finds two or more options for b with the same capacity C ((9)), it must select the one with less distortion D ((10)).

Table 1. Decision table for vector update.

Case	Condition 1	Condition 2	Condition 3	Test Bin b_t	Test Vectors a_t and b_t
1	$a_t < a_1 < b_1$				$\boldsymbol{a_t} = [a_t, a_1, \ldots, a_p]$
				$b_t = minb(0, a_1 - 1, a_t)$	$\boldsymbol{b_t} = [b_t, b_1, \ldots, b_p]$
2	$b_p < a_p < a_t$				$\boldsymbol{a_t} = [a_1, \ldots, a_p, a_t]$
				$b_t = minb(a_p + 1, G, a_t)$	$\boldsymbol{b_t} = [b_1, \ldots, b_p, b_t]$
3	$b_s < a_s < a_t < a_{s+1} < b_{s+1}$				$\boldsymbol{a_t} = [a_1, \ldots, a_s, a_t, a_{s+1}, \ldots, a_p]$
				$b_t = minb(a_s + 1, a_{s+1} - 1, a_t)$	$\boldsymbol{b_t} = [b_1, \ldots, b_s, b_t, b_{s+1}, \ldots, b_p]$
4	$a_t < b_1 < a_1$	$a_1 - a_t > 2$			$\boldsymbol{a_t} = [a_t, a_1, \ldots, a_p]$
			$\exists\, b_2$	$b_t = minb(b_1 + 2, \min(a_2, b_2) - 1, a_1)$	$\boldsymbol{b_t} = [b_1, b_t, b_2, \ldots, b_p]$
			$\nexists\, b_2$	$b_t = minb(b_1 + 2, G, a_1)$	$\boldsymbol{b_t} = [b_1, b_t]$
		$a_1 - a_t \leq 2$			$\boldsymbol{a_t} = [a_t, a_1, \ldots, a_p]$
				$b_t = minb(0, b_1 - 1, a_t)$	$\boldsymbol{b_t} = [b_t, b_1, \ldots, b_p]$
5	$a_p < b_p < a_t$	$a_t - a_p > 2$			$\boldsymbol{a_t} = [a_1, \ldots, a_p, a_t]$
			$\exists\, b_{p-1}$	$b_t = minb(\max(b_{p-1}, a_{p-1}) + 1, b_p - 2, a_p)$	$\boldsymbol{b_t} = [b_1, \ldots, b_{p-1}, b_t, b_p]$
			$\nexists\, b_{p-1}$	$b_t = minb(0, b_p - 2, a_p)$	$\boldsymbol{b_t} = [b_t, b_p]$
		$a_t - a_p \leq 2$			$\boldsymbol{a_t} = [a_1, \ldots, a_p, a_t]$
				$b_t = minb(b_p + 1, G, a_t)$	$\boldsymbol{b_t} = [b_1, \ldots, b_p, b_t]$
6	$a_s < b_s < a_t < a_{s+1} < b_{s+1}$	$a_t - a_s > 2$			$\boldsymbol{a_t} = [a_1, \ldots, a_s, a_t, a_{s+1}, \ldots, a_p]$
			$\exists\, b_{s-1}$	$b_t = minb(\max(a_{s-1}, b_{s-1}) + 1, b_s - 2, a_s)$	$\boldsymbol{b_t} = [b_1, \ldots, b_{s-1}, b_t, b_s, b_{s+1}, \ldots, b_p]$
			$\nexists\, b_{s-1}$	$b_t = minb(0, b_s - 2, a_s)$	$\boldsymbol{b_t} = [b_t, b_s, b_{s+1}, \ldots, b_p]$
		$a_t - a_s \leq 2$			$\boldsymbol{a_t} = [a_1, \ldots, a_s, a_t, a_{s+1}, \ldots, a_p]$
				$b_t = minb(b_s + 1, a_{s+1} - 1, a_t)$	$\boldsymbol{b_t} = [b_1, \ldots, b_s, b_t, b_{s+1}, \ldots, b_p]$
7	$b_s < a_s < a_t < b_{s+1} < a_{s+1}$	$a_{s+1} - a_t > 2$			$\boldsymbol{a_t} = [a_1, \ldots, a_s, a_t, a_{s+1}, \ldots, a_p]$
			$\exists\, b_{s+2}$	$b_t = minb(b_{s+1} + 2, \min(b_{s+2}, a_{s+2}) - 1, a_{s+1})$	$\boldsymbol{b_t} = [b_1, \ldots, b_s, b_{s+1}, b_t, b_{s+2}, \ldots, b_p]$
			$\nexists\, b_{s+2}$	$b_t = minb(b_{s+1} + 2, G, a_{s+1})$	$\boldsymbol{b_t} = [b_1, \ldots, b_s, b_{s+1}, b_t]$
		$a_{s+1} - a_t \leq 2$			$\boldsymbol{a_t} = [a_1, \ldots, a_s, a_t, a_{s+1}, \ldots, a_p]$
				$b_t = minb(a_s + 1, b_{s+1} - 1, a_t)$	$\boldsymbol{b_t} = [b_1, \ldots, b_s, b_t, b_{s+1}, \ldots, b_p]$

Table 1. *Cont.*

Case	Condition 1	Condition 2	Condition 3	Test Bin b_t	Test Vectors $\boldsymbol{a_t}$ and $\boldsymbol{b_t}$
8	$a_s < b_s < a_t < b_{s+1} < a_{s+1}$	$a_t - a_s > 2$			$\boldsymbol{a_t} = [a_1, \ldots, a_s, a_t, a_{s+1}, \ldots, a_p]$
			$\exists\, b_{s-1}$	$b_t = minb(\max(b_{s-1}, a_{s-1}) + 1, b_s - 2, a_s)$	$\boldsymbol{b_t} = [b_1, \ldots, b_{s-1}, b_t, b_s - 1, b_{s+1}, \ldots, b_p]$
			$\nexists\, b_{s-1}$	$b_t = minb(0, b_s - 2, a_s)$	$\boldsymbol{b_t} = [b_t, b_s - 1, b_{s+1}, \ldots, b_p]$
		$a_{s+1} - a_t > 2$			$\boldsymbol{a_t} = [a_1, \ldots, a_s, a_t, a_{s+1}, \ldots, a_p]$
			$\exists\, b_{s+2}$	$b_t = minb(b_{s+1} + 2, \min(b_{s+2}, a_{s+2}) - 1, a_{s+1})$	$\boldsymbol{b_t} = [b_1, \ldots, b_s, b_{s+1}, b_t, b_{s+2}, \ldots, b_p]$
			$\nexists\, b_{s+2}$	$b_t = minb(b_{s+1} + 2, G, a_{s+1})$	$\boldsymbol{b_t} = [b_1, \ldots, b_s, b_{s+1}, b_t]$
					$\boldsymbol{a_t} = [a_1, \ldots, a_s, a_t, a_{s+1}, \ldots, a_p]$
				$b_t = minb(b_s + 1, b_{s+1} - 1, a_t)$	$\boldsymbol{b_t} = [b_1, \ldots, b_s, b_t, b_{s+1}, \ldots, b_p]$
9	$b_s < a_t < a_s$	$a_t - b_s > 1$	$\exists\, a_{s-1}$		$\boldsymbol{a_t} = [a_1, \ldots, a_{s-1}, a_t, a_s, \ldots, a_p]$
			$\nexists\, a_{s-1}$		$\boldsymbol{a_t} = [a_t, a_s, \ldots, a_p]$
			$\exists\, b_{s+1}$	$b_t = minb(a_t + 1, \min(b_{s+1}, a_{s+1}) - 1, a_s)$	$\boldsymbol{b_t} = [b_1, \ldots, b_s, b_t, b_{s+1}, \ldots b_p]$
			$\nexists\, b_{s+1}$	$b_t = minb(a_t + 1, G, a_s)$	$\boldsymbol{b_t} = [b_1, \ldots, b_s, b_t]$
10	$a_s < a_t < b_s$	$b_s - a_t > 1$	$\exists\, a_{s+1}$		$\boldsymbol{a_t} = [a_1, \ldots, a_s, a_t, a_{s+1}, \ldots, a_p]$
			$\nexists\, a_{s+1}$		$\boldsymbol{a_t} = [a_1, \ldots, a_s, a_t]$
			$\exists\, b_{s-1}$	$b_t = minb(\max(b_{s-1}, a_{s-1}) + 1, a_t - 1, a_s)$	$\boldsymbol{b_t} = [b_1, \ldots, b_{s-1}, b_t, b_s, \ldots, b_p]$
			$\nexists\, b_{s-1}$	$b_t = minb(0, a_t - 1, a_s)$	$\boldsymbol{b_t} = [b_t, b_s, \ldots, b_p]$

3.3. Generation and Reading of Removal Sequences

As mentioned above, the information to extract the hidden data and recover the original image is contained in the internal and user removal sequences. The user removal sequence k contains the information to remove the internal removal sequence ki and subsequently is used to remove the hidden information BS from the enhanced image. Section 3.3.1 explains the generation process of the internal removal sequence ki, and Section 3.3.2 describes how to recover the information from ki.

3.3.1. Generation of Internal and User Removal Sequence

For the sake of brevity, we explain the process to generate internal removal sequence ki; however, note that this procedure also applies to building the user removal sequence k.

The removal sequence ki is composed by the concatenation of:

1. (sep): Predefined flag to separate main parameters;
2. $(sepr1, sepr2, \ldots)$: Flags to separate the peak and the minimum bins;
3. (F), (C): Block division in rows and columns;
4. (L): Number of embedding levels;
5. (G): Length of the binary string BS;
6. $(a11, a12, a13, \ldots a21, a22, a23, \ldots)$: All peak bins;
7. $(b11, b12, b13, \ldots b21, b22, b23, \ldots)$: All minimum bins;
8. $(nbran)$: Total bits to represent the number of ranges per block;
9. (nba): Total bits to represent the peak bins;
10. (nbb): Total bits to represent the minimum bins.

The flag sep is a user-selected value that delimits the parameters into the removal sequence ki, used for reversibility. The flags $sepr1, sepr2, \ldots$ contain the number of ranges appointed to each block, number $1, 2, \ldots$, respectively. The parameter $nbran$ is in the function of the maximum possible number of ranges $maxNR$ contained in each block, $nbran = \lceil \log_2(maxNR + 1) \rceil$, where $\lceil \cdot \rceil$ is the nearest upper integer. Values nba and nbb are obtained using the maximum values $maxa$ and $maxb$ in $\boldsymbol{a}$ and $\boldsymbol{b}$, respectively, determined by $nba = \lceil \log_2(maxa + 1) \rceil$ and $nbb = \lceil \log_2(maxb + 1) \rceil$.

The peak bins $(a11, a12, a13, \ldots a21, a22, a23, \ldots)$ and the minimum bins $(b11, b12, b13, \ldots b21, b22, b23, \ldots)$ are the values contained in vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, whose notation is given by the letters a and b followed by the number of the range as well as the number of the level, respectively, e.g., $a23$ represents the peak bin of the range number 2 in the enhanced image of level L = 3.

The order of the data into the removal sequence ki is given by (18).

$$\begin{aligned} ki = [&F, sep, C, sep, L, sep, G, sep, nbran, sep, nba, sep, nbb, sep, \\ &sepr1, a11, a12, a13, \ldots, b11, b12, b13, \ldots, \\ &sepr2, a21, a22, a23, \ldots, b21, b22, b23, \ldots], \end{aligned} \tag{17}$$

3.3.2. Reading of Internal and User Removal Sequence

The internal removal sequence ki is segmented by detecting the location of the flag values sep contained within the binary string. The first seven segments between sep values correspond to F, C, L, G, $nbran$, nba, and nbb according to (17). Subsequently, we read the next $nbran$ number of bits, and the read value corresponds to $sepr1$. We read the next nba bits consecutively $sepr1$ times, and the next nbb bits $sepr1$ times to extract the peak and minimum bins, respectively, as shown in (17). Again, we read the next $nbran$ number of bits to read the value corresponding to $sepr2$, and the process is repeated until the end of the string. The extracted peak bins $a11, a12, a13, \ldots a21, a22, a23, \ldots$ and minimum bins $b11, b12, b13, \ldots b21, b22, b23, \ldots$ are the elements of the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ in the corresponding block and level.

4. Experimental Results

Experimental results were performed on USC-SIPI [32] dataset composed of eight images of 512 × 512 in size, Kodak Lossless True Color Image Suite [33] that contain 24 images of 768 × 512 in size, and BOWS-2 dataset [34] with five images of 512 × 512 in size, all images have 8-bit depth grayscale. We conducted several experiments to evaluate the proposed algorithm in terms of capacity and visual quality. First, we presented the assessment metrics and parameter settings. Then, we evaluated and discussed the results on three common individual images. Finally, we presented a comparison performance with eight previous methods reported in the state-of-the-art proposals [21,24–26,28–31], showing that our proposal outperforms all of them and gives new visual quality benefits.

4.1. Assessment Metrics to Contrast-Enhancement and Visual Quality

The goal of the data hiding schemes with contrast enhancement is to embed information and, at the same time, enhance the contrast of the image. To assess the contrast increment of the enhanced image, we used the relative entropy error (REE), the relative mean brightness error (RMBE), and the relative contrast error (RCE) [27]. REE is in the range [0, 1], where REE > 0.5 means that the entropy of the image has been increased. For the user, the RMBE metric measures the change of the brightness in the image, RMBE = 1 indicates that the brightness has not been changed, RMBE < 1 indicates that the brightness has been altered. Finally, RCE measures the contrast change, and RCE > 0.5 shows an increment in the contrast of the enhanced image. To evaluate the visual quality of the image, we used a nonreferenced metric called Blind/Reference-Less Image Quality Evaluator (BRISQUE) [35], which measures the damage caused by noise and blurring into the images without any reference. Finally, we employed the well-known reference quality metrics called Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index (SSIM) [36], which are applied to the enhanced image with respect to the original image.

4.2. Parameters Settings

Based on several experiments, we recommend the set of parameters shown in Table 2, which are used in the experimental results.

Table 2. Summary of recommended parameters.

	BS **Embedding**	ki **Embedding**
sep (delimiter flag)	1011001110001111	1011001110001111
Vp (stop value)	0.05	0.05
$F = C$	1, 2, 4, 8	1

The parameters that most affect the performance are F and C and can be selected according to the application. In Figure 9, we hide 0.8 bpp in the Lena image and showed how $F \times C$ number of blocks, when hiding the binary string BS, affect quality and processing time, considering $F = C$. The three black curves correspond to different values of F (2, 4, and 8) when hiding the binary string BS. Additionally, the red lines represent the quality decrement when hiding the internal removal sequence with $F = 1$ after hiding BS. Finally, each marker is the transition between embedding levels.

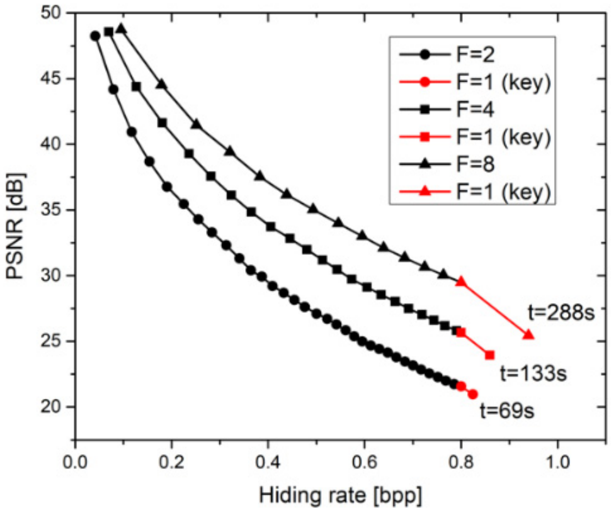


Figure 9. PSNR visual quality according to hiding rate.

We can see in Figure 9 that if we define $F = 8$ for *BS* embedding, the quality is higher, but the processing time *t* increases. On the other hand, if we define $F = 2$, the quality is less, and the embedding process is faster. Additionally, the level 'markers' for $F = 2$ are closer, which means the capacity at each level decreases faster. This might cause the maximum capacity to be reached before embedding all the information, and therefore, we do not recommend a small number of blocks for higher payloads. Since most RDH-CE consider quality more important than processing time, the results for comparison are obtained using $F = 8$.

Table 3 shows different qualitative results using different values of *F*. We can see that the lengths of the removal sequences *ki* and *k* increase when *F* and *C* are increased. However, the length of the user removal sequence *k* remains under 5% of the length of the internal removal sequence *ki*, which proves the effectiveness of the internal removal sequence hiding process.

Table 3. Quantitative results.

Inputs	F, C for *BS*	2	4	8
	F, C for *ki*		1	
	BS length (0.8 bpp) (Bytes)		26214	
Outputs	*ki* length (Bytes)	795	1953	4565
	k length (Bytes)	36	51	75
	Embedding time (s)	69	133	288
	Embedding levels for *BS*	34	22	14
	Embedding levels for *ki*	2	4	8
	RCE	0.580	0.549	0.539
	PSNR (dB)	20.96	25.66	29.49
	SSIM	0.817	0.828	0.823

4.3. Performance on Individual Images

One condition that cannot be controlled in advance and may affect the performance is the histograms distribution of the original image. Figure 10a–c shows F-16, Lena, and Baboon images, respectively, with a block rounded in red. Figure 10d–f shows, with a black line, the distribution of the histogram in the selected squares that corresponds to each image. Additionally, the red lines correspond to the distributions after embedding the information with the present RDH-CE proposal. Since the three images have different sharpness in the original histogram, we use them to test performance.

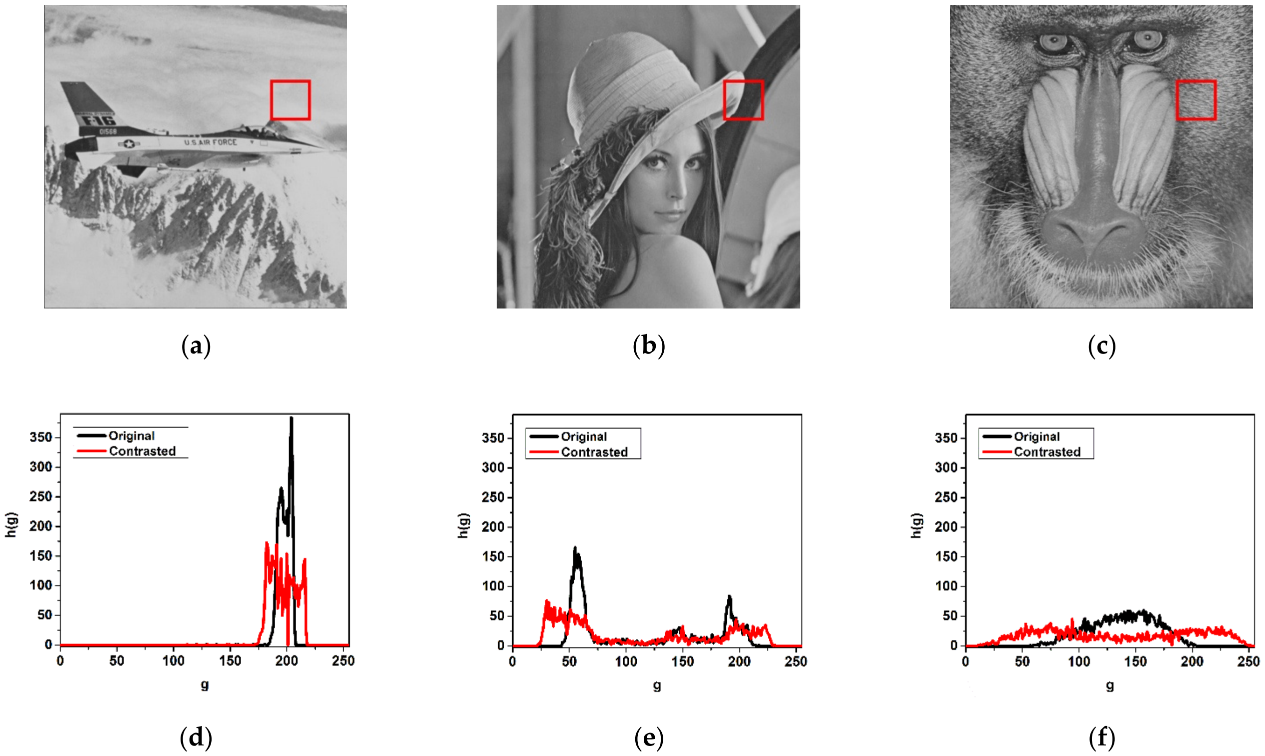
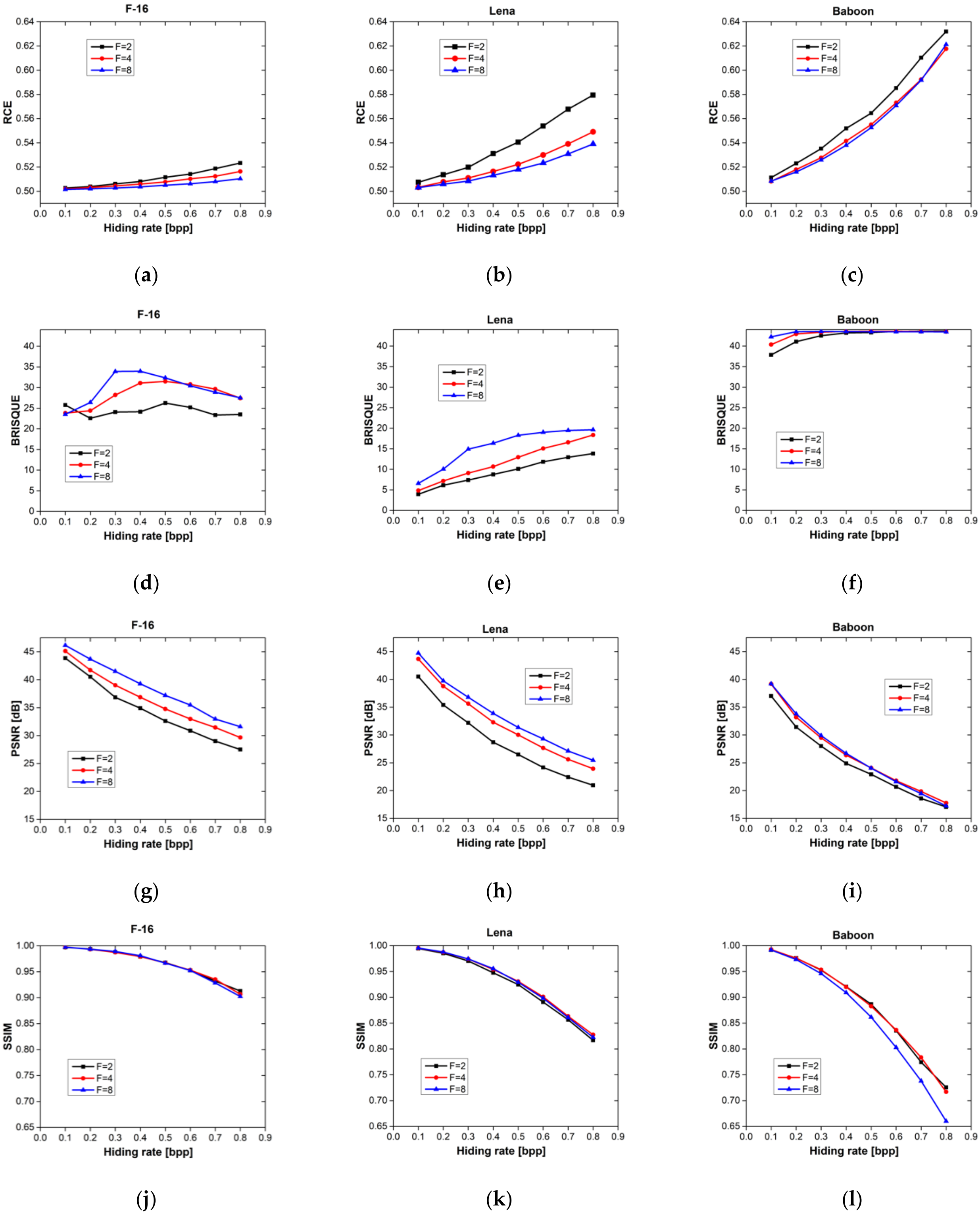


Figure 10. Original images: (**a**) F-16; (**b**) Lena; (**c**) Baboon with a selected block (red square); (**d**–**f**) original histogram (black line) and enhanced histogram (red line) of the selected block.

Figure 11 shows the contrast RCE, the nonreferenced quality BRISQUE, and the referenced qualities PSNR and SSIM applied to the images in Figure 10a–c after hiding information. The size of the embedded payload goes from 0.1 to 0.8 bpp of information. We can see in Figure 11a,d,g,j, F-16 image with the sharpest histograms is not considerably altered. This is because we have more capacity given by higher peak bins when hiding information. On the other hand, Figure 11b,e,h,k show higher image modification after embedding information. Finally, Figure 11c,f,i,l shows the results for the Baboon image, which is considerably altered. This is because the peak bins are smaller, and the algorithm needs more levels to hide the information.

Figure 12 shows the visual results of the three images considering a fixed hiding rate of 0.8 bpp with $F = 8$ for BS embedding and $F = 1$ for embedding the removal sequence ki. From left to right, the first column in Figure 12 shows the original images and the initial RCE, BRISQUE, PSNR, and SSIM values. The second column shows the results for the images with the BS string hidden, and the third column shows the results for the image with the BS and ki data concealed. We can see how the contrast is increased.



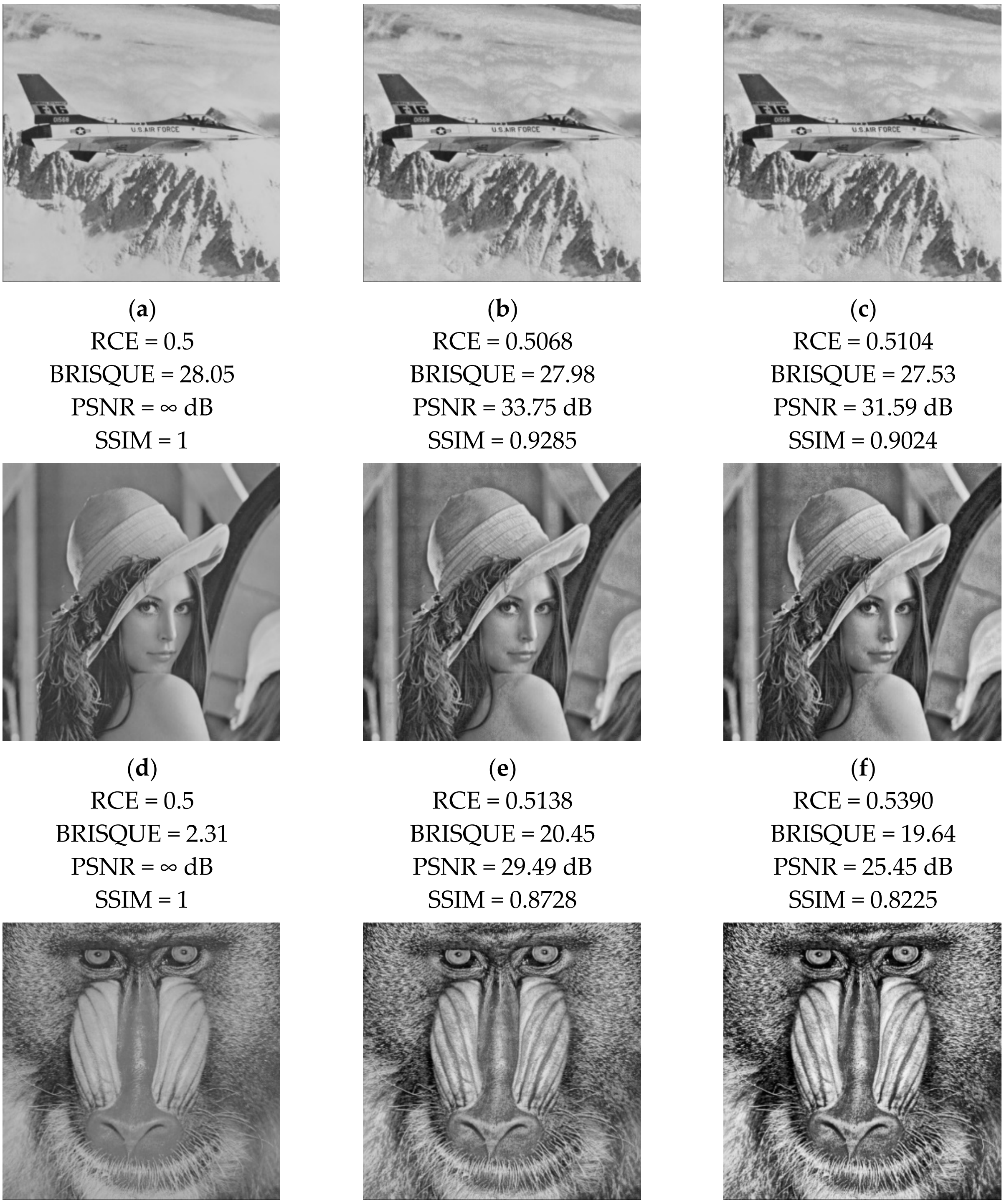
(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

(**g**) (**h**) (**i**)

(**j**) (**k**) (**l**)

Figure 11. Assessment metrics with different hiding rates for F-16, Lena, and Baboon images: RCE (**a**–**c**); BRISQUE (**d**–**f**); PSNR (**g**–**i**); SSIM (**j**–**l**).



(**a**)
RCE = 0.5
BRISQUE = 28.05
PSNR = ∞ dB
SSIM = 1

(**b**)
RCE = 0.5068
BRISQUE = 27.98
PSNR = 33.75 dB
SSIM = 0.9285

(**c**)
RCE = 0.5104
BRISQUE = 27.53
PSNR = 31.59 dB
SSIM = 0.9024

(**d**)
RCE = 0.5
BRISQUE = 2.31
PSNR = ∞ dB
SSIM = 1

(**e**)
RCE = 0.5138
BRISQUE = 20.45
PSNR = 29.49 dB
SSIM = 0.8728

(**f**)
RCE = 0.5390
BRISQUE = 19.64
PSNR = 25.45 dB
SSIM = 0.8225

(**g**)
RCE = 0.5
BRISQUE = 32.7589
PSNR = ∞ dB
SSIM = 1

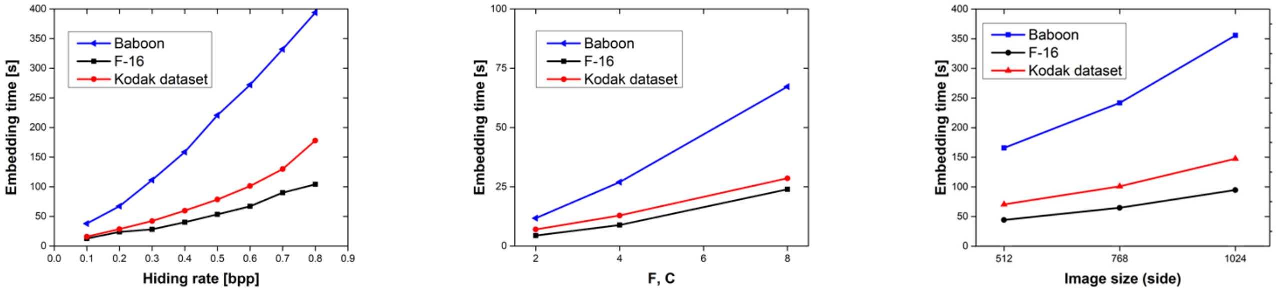
(**h**)
RCE = 0.5529
BRISQUE = 43.50
PSNR = 22.83 dB
SSIM = 0.8167

(**i**)
RCE = 0.6212
BRISQUE = 43.46
PSNR = 17.23 dB
SSIM = 0.6600

Figure 12. (**a**,**d**,**g**) Original images; (**b**,**e**,**h**) enhanced images with internal removal sequence *ki* embedded; (**c**,**f**,**i**) enhanced images with internal removal sequence *ki* and binary string *BS* embedded.

4.4. Computational Complexity (Speed)

Our method has been implemented using MATLAB© R2018a on a Notebook with an Intel© i5-8250U processor (1.6 GHz) and 8 GB random access memory (RAM). The three main parameters that may affect processing time are hiding rate, number of blocks, and image size. Additionally, one factor that considerably affects the speed of the embedding and removal procedures is histogram distribution; however, the latter depends on image content, and we cannot control it. Figures 13 and 14 show three curves that correspond to the F-16 and Baboon images (size 512 × 512) shown in Figure 10a,c, and a curve which is the average results of the 24 images in the Kodak dataset (resized to 512 × 512). In all graphs, Baboon shows to have the highest processing times since it has high distributed histograms with reduced initial capacity, as shown in Figure 10f; consequently, the algorithm needs more iterations and time to hide the information. In contrast, the F-16 image contains sharper histograms, as shown in Figure 10d, and therefore, the processing is faster.



(**a**) (**b**) (**c**)

Figure 13. Embedding time according to (**a**) hiding rate with $F = C = 8$, (**b**) number of blocks ($F \times C$) for a hiding rate equal to 0.2 bpp, and (**c**) image size with a hiding rate of 0.5 bpp and $F = C = 8$.

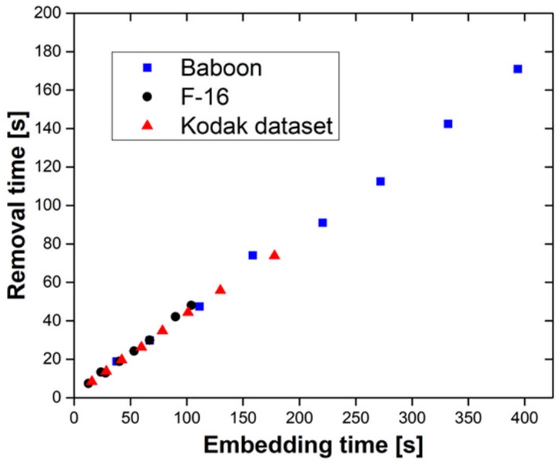


Figure 14. Relation between embedding time and removal time using Figure 13a information.

Figure 13a shows the embedding time according to the hiding rate; as expected, time increases when we raise the length of the concealed information. Figure 13b shows that if we increase the number of blocks ($F \times C$), the embedding time also increases, and, as previously discussed in Figure 9, the quality of the image may be improved. To measure how the image size affects processing time, Figure 13c shows the case in which the hiding rate is 0.5 bpp; to maintain this value constant, we must increase the length of the embedded information, and consequently, time increases too.

Finally, Figure 14 shows a sparse graph that compares embedding time with the removal time of the points in Figure 13a. The relation seems to be linear and independent of the histogram distribution between images. The experiment shows that the removal process lasts approximately 0.46 times the embedding stage, on average.

4.5. Performance Comparison

Comparisons of performance between our proposal and those most recent schemes [21,24–26,28–31] are shown in Tables 4–8. Values in the grids are highlighted in bold style to indicate they outperform in the comparison. The proposals in [21,24–26,28–31] are based on a preprocessing that determines S shifting iterations to conceal the data bits. For a fair comparison, our proposal embeds a similar hiding rate to that obtained by different values of S. The metrics used in the results are described in Section 4.1.

Table 4. Average metrics with USC-SIP dataset using $F, C = 8$ for BS embedding.

S	Hiding Rate	Method	RCE	REE	RMBE	BRSIQUE	SSIM	PSNR (dB)
20	0.458	[21]	**0.548**	0.526	0.989	23.54	0.920	25.05
	0.451	[24]	0.547	0.525	0.989	23.41	0.927	25.09
	0.482	[25]	0.537	0.525	**0.977**	**22.19**	**0.943**	26.43
	0.452	[29]	0.543	0.526	0.987	22.74	0.936	25.54
	0.447	[31]	0.545	**0.530**	0.990	22.64	0.932	25.45
	0.450	Proposed	0.518	0.515	0.997	30.13	0.923	**31.99**
30	0.607	[21]	**0.567**	0.533	0.983	27.89	0.852	21.72
	0.590	[24]	0.565	0.532	0.983	27.00	0.860	22.01
	0.582	[29]	0.555	0.533	**0.980**	23.48	**0.903**	23.27
	0.581	[31]	0.561	**0.538**	0.985	**23.47**	0.893	22.94
	0.600	Proposed	0.527	0.521	0.995	30.16	0.878	**28.85**
50	0.865	[21]	**0.581**	**0.537**	0.966	44.38	0.647	17.28
	0.835	[24]	0.571	0.535	**0.952**	**40.56**	0.675	17.44
	0.850	Proposed	0.555	0.532	0.990	32.09	**0.755**	**23.68**
	0.676	[29]	0.572	0.538	0.967	**24.30**	**0.853**	20.76
	0.700	[31]	**0.583**	**0.546**	**0.962**	24.69	0.822	19.69
	0.675	Proposed	0.534	0.524	0.994	30.57	0.840	**27.18**

Table 5. Average metrics with Kodak dataset using $F, C = 8$ for BS embedding.

S	Hiding Rate	Method	RCE	REE	RMBE	BRSIQUE	SSIM	PSNR (dB)
20	0.511	[21]	**0.544**	0.528	0.985	16.65	0.902	24.89
	0.496	[24]	0.539	0.529	0.984	15.42	0.914	24.70
	0.563	[25]	0.541	0.532	0.966	14.01	0.915	23.37
	0.476	[29]	0.537	0.528	**0.982**	14.00	0.928	25.02
	0.518	[31]	0.539	**0.534**	**0.982**	**13.79**	0.921	24.85
	0.500	Proposed	0.513	0.515	0.997	25.72	**0.947**	**34.43**
30	0.695	[21]	**0.560**	0.535	0.976	20.57	0.831	21.34
	0.655	[24]	0.555	0.535	0.975	19.84	0.843	21.53
	0.564	[29]	0.550	0.533	**0.970**	**15.08**	0.890	22.27
	0.639	[31]	0.555	**0.541**	0.974	15.59	0.871	21.74
	0.638	Proposed	0.520	0.520	0.996	25.65	**0.913**	**31.84**
50	0.847	[21]	**0.581**	**0.537**	0.966	**44.38**	0.647	17.28
	0.800	[24]	0.571	0.535	**0.952**	40.56	0.675	17.44
	0.817	Proposed	0.550	0.530	0.991	31.80	**0.775**	**24.400**
	0.603	[29]	0.572	0.538	0.967	**24.30**	0.853	20.76
	0.715	[31]	**0.583**	**0.546**	**0.962**	24.69	0.822	19.69
	0.638	Proposed	0.520	0.520	0.996	25.65	**0.913**	**31.84**

Table 6. Average metrics with Kodak dataset using $F, C = 1$ for BS embedding.

S Value	Hiding Rate	Method	RCE	REE	RMBE	BRSIQUE	SSIM	PSNR (dB)
20	0.511	[21]	0.544	0.528	0.985	16.65	0.902	24.89
	0.496	[24]	0.539	0.529	0.984	15.42	0.914	24.70
	0.563	[25]	0.541	0.532	**0.966**	14.01	0.915	23.37
	0.476	[29]	0.537	0.528	0.982	14.00	0.928	25.02
	0.518	[31]	0.539	**0.534**	0.982	13.79	0.921	24.85
	0.510	Proposed	**0.547**	0.533	0.987	**9.75**	**0.930**	**25.74**
30	0.695	[20]	0.560	0.535	0.976	20.57	0.831	21.34
	0.655	[24]	0.555	0.535	0.975	19.84	0.843	21.53
	0.564	[29]	0.550	0.533	**0.970**	15.08	**0.890**	22.27
	0.639	[31]	0.555	**0.541**	0.974	15.59	0.871	21.74
	0.638	Proposed	**0.563**	**0.541**	0.975	**12.21**	**0.890**	**22.97**

Table 7. Average metrics using the images of Figure 15 $F, C = 8$.

Hiding Rate	Method	RCE	REE	RMBE	SSIM	PSNR
0.708	[30]	0.553	**0.536**	**0.958**	0.871	22.99
0.670	[21]	**0.592**	0.528	0.960	0.828	22.58
0.699	[26]	0.555	0.532	0.968	0.858	23.34
0.691	[28]	0.557	0.530	0.968	0.860	22.53
0.691	Proposed	0.518	0.527	0.997	**0.915**	**32.40**

Table 8. Average metrics using the images of Figure 15 and $F, C = 1$.

Hiding Rate	Method	RCE	REE	RMBE	SSIM	PSNR
0.708	[30]	0.553	0.536	**0.958**	0.871	22.99
0.670	[21]	**0.592**	0.528	0.960	0.828	22.58
0.699	[26]	0.555	0.532	0.968	0.858	23.34
0.691	[28]	0.557	0.530	0.968	0.860	22.53
0.691	Proposed	0.543	**0.541**	0.979	**0.900**	**24.34**

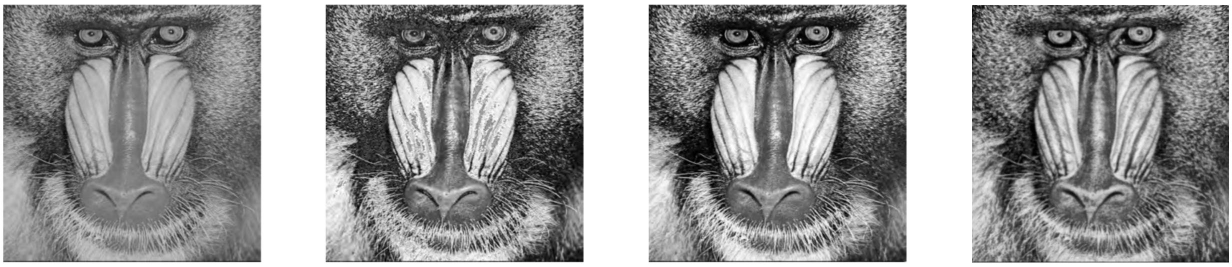
Table 4 reports experimental results on average for the USC-SIP dataset and Table 5 for the Kodak dataset. Additionally, we consider the recommended parameters of Table 4 and define $F = C = 8$ for BS embedding. We can see that values RCE, REE, and RMBE for our proposal are close to those obtained in other schemes; nevertheless, the proposed method enhances the contrast differently, as shown later. The values of the nonreferenced quality metric BRISQUE are the worst in our proposal because the algorithm amplifies noises that are unseen in the original image. As shown in Table 4, the quality SSIM and PSNR are less but similar for the USC-SIP database, which contains eight test images. However, if we increase to 24 images using the Kodak dataset, we can see in Table 5 that our SSIM and PSNR values increase and become better than in other approaches.

If we want to increase contrast and maintain the same hiding rate, we can decrease the number of blocks (with F, C), but the capacity is more restricted, as discussed in Section 4.2. However, Table 6 shows results obtained using the Kodak dataset and recommended block division from Table 2 as $F = C = 1$. We can see that contrast becomes closer to other proposals and, in some cases, is higher. SSIM and PSNR values continue being the best even with the number of blocks decreasing.

Considering a set of five test images shown in Figure 15, a performance comparison in terms of average RCE, REE, RMBE, SSIM, and PSNR, with schemes from [19,24,26,28] is shown in Tables 7 and 8, using $F = C = 8$ and $F = C = 1$ for the BS embedding, respectively. From Tables 7 and 8, we show that the metrics RCE, REE, and RMBE indicate less contrast obtained by our proposal when $F, C = 8$ and a gained contrast revealed by the metric REE when $F, C = 1$. On the other hand, in all cases, the values of the SSIM and PSNR metrics obtained by our proposal outperform the algorithms in [24,26,28,29], resulting in a good visual quality of enhanced images.

Figure 15. Test images used in the experimental results of Tables 7 and 8.

For illustrative purposes, in Figures 16–18, we can see a visual comparison between schemes [21,31] and the proposed method, using a couple of images from USC-SIP and Kodak datasets, respectively. From Figures 16 and 17, we show that methods [21,31] and our proposal obtain a similar value of RCE in all cases. On the other hand, the scheme reported in [21] and the proposed method outperforms [31] in terms of BRISQUE, both keeping a consistent value of this metric. However, the proposed method has several advantages over [21,31] in terms of visual quality and capacity. By a naked eye, from Figures 16 and 17, we show that our method enhances the contrast of the images obtaining a sharp definition to the edges, contours, and textures without damaging the luminance in dark and bright areas, compared with the results obtained by [21,31]. In this context, by a naked eye, we can see that the scheme [21] causes false contours in the images. For the user, the method in [31] does not reveal content in the regions with low and high luminance, as shown in Figure 18.



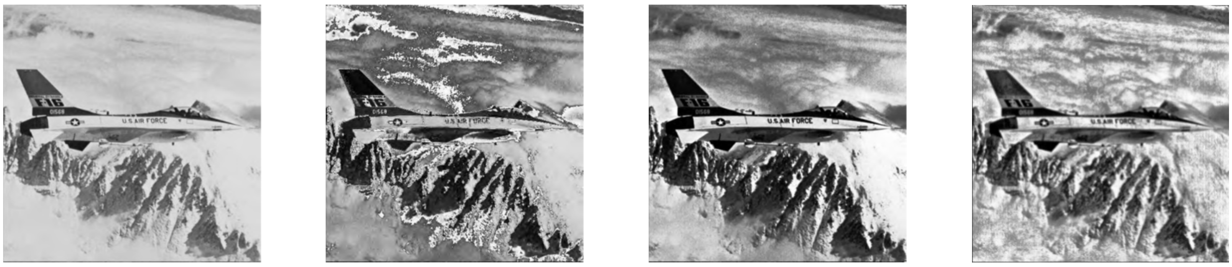
(**a**)

(**b**)
Enhanced image by Wu, H.T., et al. 2014
(S = 64)
RCE = 0.631
PSNR = 14.85 dB
SSIM = 0.487
Hiding rate = 0.732 bpp
BRISQUE = 55.84

(**c**)
Enhanced image by Wu, H.T., et al. 2019
(S = 64)
RCE = 0.623
PSNR = 17.05 dB
SSIM = 0.679
Hiding rate = 0.545 bpp
BRISQUE = 24.34

(**d**)
Enhanced by the proposed method
RCE = 0.621
PSNR = 17.23 dB
SSIM = 0.660
Hiding rate = 0.8 bpp
BRISQUE = 43.46

Figure 16. *Cont.*



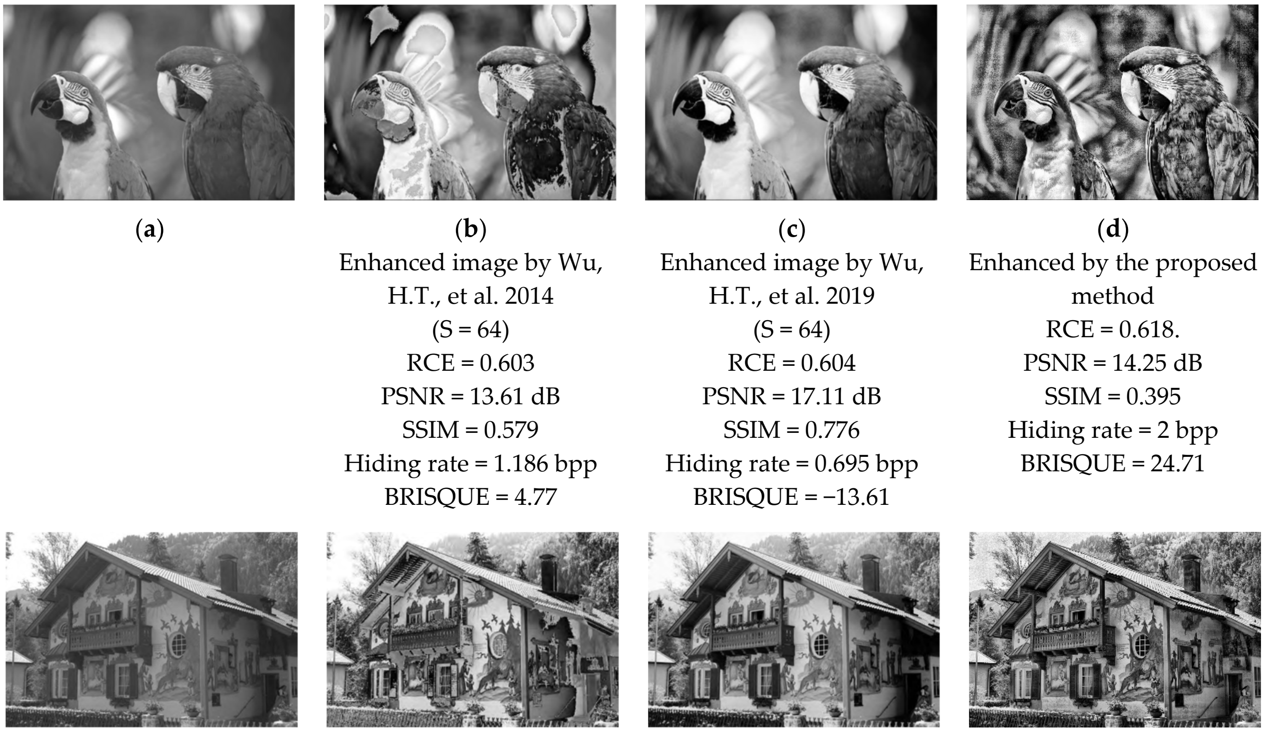
(**e**)

(**f**)
Enhanced image by Wu, H.T., et al. 2014
(S = 64)
RCE = 0.591
PSNR = 11.29 dB
SSIM = 0.405
Hiding rate = 1.537 bpp
BRISQUE = 37.01

(**g**)
Enhanced image by Wu, H.T., et al. 2019
(S = 64)
RCE = 0.594
PSNR = 13.12 dB
SSIM = 0.585
Hiding rate = 1.227 bpp
BRISQUE = 6.91

(**h**)
Enhanced by the proposed method
RCE = 0.598
PSNR = 16.71 dB
SSIM = 0.477
Hiding rate = 1.8 bpp
BRISQUE = 33.04

Figure 16. Two original images of USC-SIP dataset (**a**) Baboon and (**e**) F-16. Enhanced images obtained by algorithms in (**b**,**f**) [21], (**c**,**g**) [31], and (**d**,**h**) the present proposal.



(**a**)

(**b**)
Enhanced image by Wu, H.T., et al. 2014
(S = 64)
RCE = 0.603
PSNR = 13.61 dB
SSIM = 0.579
Hiding rate = 1.186 bpp
BRISQUE = 4.77

(**c**)
Enhanced image by Wu, H.T., et al. 2019
(S = 64)
RCE = 0.604
PSNR = 17.11 dB
SSIM = 0.776
Hiding rate = 0.695 bpp
BRISQUE = −13.61

(**d**)
Enhanced by the proposed method
RCE = 0.618.
PSNR = 14.25 dB
SSIM = 0.395
Hiding rate = 2 bpp
BRISQUE = 24.71

(**e**)

(**f**)
Enhanced image by Wu, H.T., et al. 2014
(S = 64)
RCE = 0.580
PSNR = 13.32 dB
SSIM = 0.411
Hiding rate = 1.042 bpp
BRISQUE = 50.71

(**g**)
Enhanced image by Wu, H.T., et al. 2019
(S = 64)
RCE = 0.572
PSNR = 17.81 dB
SSIM = 0.686
Hiding rate = 0.808 bpp
BRISQUE = 49.76

(**h**)
Enhanced by the proposed method
RCE = 0.580
PSNR = 17.51 dB
SSIM = 0.580
Hiding rate = 1.4 bpp
BRISQUE = 43.27

Figure 17. Two original images of USC-SIP dataset: (**a**) parrots and (**e**) house. Enhanced images obtained by algorithms in (**b**,**f**) [21], (**c**,**g**) [31], and (**d**,**h**) the present proposal.

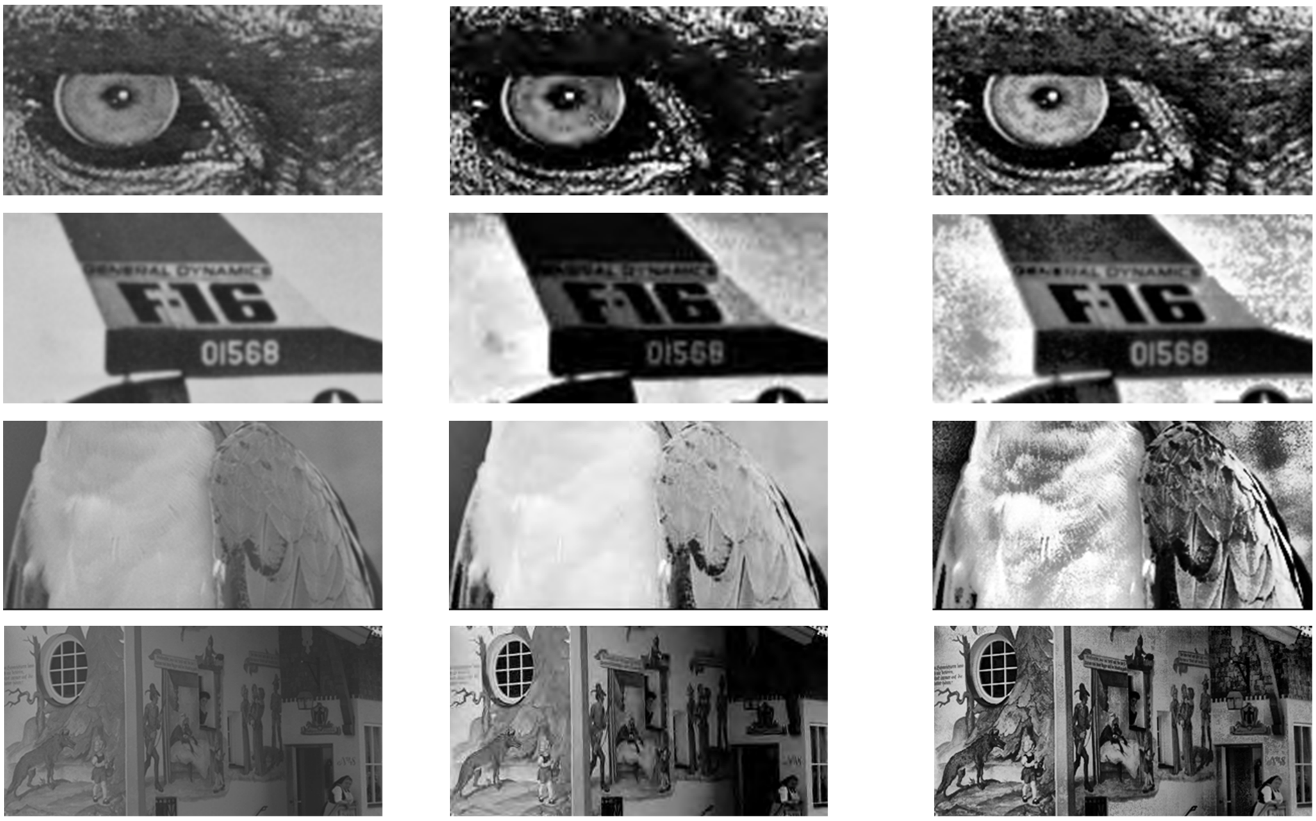


Figure 18. Zoomed parts of some original images (**left column**), enhanced images obtained by algorithm in [31] (**centered column**), and by our proposal (**right column**).

5. Discussion

Many RDH-CE algorithms attempt to increase hiding rate and image contrast simultaneously. On the other hand, the proposed approach attempts to increase the hiding rate and decrease pixel modification (distortion). Afterward, contrast increment is indirectly achieved. The result is that we conceal part of the information at each iteration while delaying the quality drop; namely, we can embed more information with less distortion to the image. To prove the previous statement, Tables 4–8 show that for a similar hiding rate, our enhanced images are less distorted. Therefore, the metrics RCE, REE, and RMBE indicate less contrast, and SSIM and PSNR indicate more quality than other published works. It seems that lower quantitative contrast is a disadvantage, but we expose three supportive arguments. Firstly, contrast metrics and visual results show that contrast is achieved. Secondly, if contrast is not enough for a specific payload, we can embed additional filling information. Finally, visual results in Figure 18 show that unseen elements of the original image are revealed better in the enhanced image than other approaches. Furthermore, PSNR less than 30 does not indicate that the image is arbitrarily distorted because contrast metrics show a contrast increment, and SSIM values demonstrate the conservation of the structure of the image.

Our visual results in Figures 16–18 show that enhanced images have more noise in regions that were plane. Therefore, we generally have the worst evaluation in the nonreferenced quality metric BRISQUE because it measures the noise increment as a negative feature. However, this noise is not added by the algorithm but is part of the original image. Furthermore, we can indicate two advantages of the proposal. Firstly, the proposed method not just increases noise (Figure 16h) but also emphasizes textures (Figure 17d) and reveals unseen details (Figure 17d,h) better compared with previous approaches. Secondly, visual results in Figures 16–18 show that contrast is increased globally in [35] and locally in our scheme, and hence, both results can be complementary.

Therefore, the present algorithm is a novel approach to contrast images in the field of RDH-CE and can be appropriate for applications that need local contrast with an increment of details.

The parameter that most affects performance is the number of blocks defined by values F and C. If we select fewer blocks, optimization runs fewer times, and processing is faster. Additionally, the distortion decreases, but the maximum hiding rate may be reached prematurely, as shown in Figure 9. On the contrary, more blocks cause the optimization process to run more times; nevertheless, better image quality is achieved, and higher payloads can be embedded. In future work, a better optimization process will be designed to decrease processing time.

6. Conclusions

In this paper, we have proposed a novel reversible data-hiding scheme with image contrast enhancement applied to digital images. The proposed method improves the conventional histogram shifting techniques, considering several novel criteria focused on contrast enhancement and optimization of multirange selection and a multilevel embedding to outperform the quality and capacity performance obtained by the current state-of-the-art reported methods. Experimental results reveal that our method enhances the contrast of the images differently, obtaining a sharp definition of edges, contours, and textures without damaging the luminance in dark and bright areas, and at the same time allowing high hiding rates, from 0.1 up to 2 bit per pixel. Moreover, the visual quality of the images is guaranteed by the proposed algorithm, as shown in the experimentation conducted by metrics to measure the contrast increment such as REE, RMBE, RCE, and the visual distortion using the nonreferenced metric BRISQUE as well as the reference metrics PSNR and SSIM. This proposal is suitable for applications where local contrast enhancement, high payload, and reversibility should be ensured, such as medical imaging and military imagery, among others.

Author Contributions: Conceptualization, E.F.-N. and M.C.-H.; data curation, E.F.-N., M.C.-H. and F.G.-U.; formal analysis, E.F.-N., M.C.-H. and F.G.-U.; funding acquisition, M.C.-H. and F.G.-U.; investigation, E.F.-N., M.C.-H. and R.M.-Z.; methodology, E.F.-N., M.C.-H. and R.M.-Z.; project administration, M.C.-H., F.G.-U. and R.M.-Z.; resources, M.C.-H., F.G.-U. and R.M.-Z.; software, E.F.-N.; supervision, M.C.-H., F.G.-U. and R.M.-Z.; validation, M.C.-H. and F.G.-U.; writing—original draft, E.F.-N. and M.C.-H.; writing—review and editing, F.G.-U. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Universidad Nacional Autonoma de Mexico (UNAM) under the DGAPA Postdoctoral Scholarship Program, the Instituto Politecnico Nacional (IPN) via the project SIP 20220748, and the College of Engineering, San Jose State University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors thank the Universidad Nacional Autonoma de Mexico (UNAM) under the DGAPA Postdoctoral Scholarship Program, the Instituto Politecnico Nacional (IPN), and the College of Engineering, San Jose State University, for the support provided during the realization of this research. The authors of the article appreciate the valuable suggestions of the referees, which contributed to improving the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shi, Y.Q.; Li, X.; Zhang, X.; Wu, H.T.; Ma, B. Reversible Data Hiding: Advances in the Past Two Decades. *IEEE Access* **2016**, *4*, 3210–3237. [CrossRef]
2. Fridrich, J.; Goljan, M.; Du, R. Invertible Authentication. *Proc. SPIE-Int. Soc. Opt. Eng.* **2001**, *4314*, 197–208. [CrossRef]
3. Fridrich, J.; Goljan, M.; Du, R. Lossless Data Embedding-New Paradigm in Digital Watermarking. *EURASIP J. Appl. Signal Process.* **2002**, *2002*, 185–196. [CrossRef]
4. Celik, M.U.; Sharma, G.; Tekalp, A.M.; Saber, E. Lossless Generalized-LSB Data Embedding. *IEEE Trans. Image Process.* **2005**, *14*, 253–266. [CrossRef] [PubMed]
5. Tian, J. Reversible Data Embedding Using a Difference Expansion. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 890–896. [CrossRef]
6. Tian, J. Wavelet-Based Reversible Watermarking for Authentication. *Secur. Watermarking Multimed. Contents IV* **2002**, *4675*, 679–690. [CrossRef]
7. Thodi, D.M.; Rodríguez, J.J. Prediction-Error Based Reversible Watermarking. *Proc.-Int. Conf. Image Process. ICIP* **2004**, *3*, 1549–1552. [CrossRef]
8. Thodi, D.M.; Rodríguez, J.J. Expansion Embedding Techniques for Reversible Watermarking. *IEEE Trans. Image Process.* **2007**, *16*, 721–730. [CrossRef]
9. Ni, Z.; Shi, Y.Q.; Ansari, N.; Su, W. Reversible Data Hiding. *IEEE Trans. Circuits Syst. Video Technol.* **2006**, *16*, 354–361. [CrossRef]
10. Coatrieux, G.; Pan, W.; Cuppens-Boulahia, N.; Cuppens, F.; Roux, C. Reversible Watermarking Based on Invariant Image Classification and Dynamic Histogram Shifting. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 111–120. [CrossRef]
11. Lee, S.K.; Suh, Y.H.; Ho, Y.S. Reversible Image Authentication Based on Watermarking. In Proceedings of the 2006 IEEE International Conference on Multimedia and Expo, Toronto, ON, Canada, 9–12 July 2006; pp. 1321–1324. [CrossRef]
12. Li, X.; Zhang, W.; Gui, X.; Yang, B. Efficient Reversible Data Hiding Based on Multiple Histograms Modification. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 2016–2027. [CrossRef]
13. Ou, B.; Li, X.; Zhao, Y.; Ni, R. Reversible Data Hiding Using Invariant Pixel-Value-Ordering and Prediction-Error Expansion. *Signal Process. Image Commun.* **2014**, *29*, 760–772. [CrossRef]
14. Sachnev, V.; Kim, H.J.; Nam, J.; Suresh, S.; Shi, Y.Q. Reversible Watermarking Algorithm Using Sorting and Prediction. *IEEE Trans. Circuits Syst. Video Technol.* **2009**, *19*, 989–999. [CrossRef]
15. Fallahpour, M.; Sedaaghi, M.H. High Capacity Lossless Data Hiding Based on Histogram Modification. *IEICE Electron. Express* **2007**, *4*, 205–210. [CrossRef]
16. Fallahpour, M. Reversible Image Data Hiding Based on Gradient Adjusted Prediction. *IEICE Electron. Express* **2008**, *5*, 870–876. [CrossRef]
17. Hu, Y.; Lee, H.K.; Li, J. DE-Based Reversible Data Hiding with Improved Overflow Location Map. *IEEE Trans. Circuits Syst. Video Technol.* **2009**, *19*, 250–260. [CrossRef]
18. Hong, W.; Chen, T.S.; Shiu, C.W. Reversible Data Hiding for High Quality Images Using Modification of Prediction Errors. *J. Syst. Softw.* **2009**, *82*, 1833–1842. [CrossRef]
19. Hamad, S.; Khalifa, A.; Elhadad, A. A Blind High-Capacity Wavelet-Based Steganography Technique for Hiding Images into other Images. *Adv. Electr. Comput. Eng.* **2015**, *14*, 35–42. [CrossRef]
20. Elhadad, A.; Ghareeb, A.; Abbas, S. A Blind and High-Capacity Data Hiding of DICOM Medical Images Based on Fuzzification Concepts. *Alex. Eng. J. Lett.* **2021**, *60*, 2471–2482. [CrossRef]
21. Wu, H.T.; Dugelay, J.L.; Shi, Y.Q. Reversible Image Data Hiding with Contrast Enhancement. *IEEE Signal Process. Lett.* **2015**, *22*, 81–85. [CrossRef]
22. Stark, J.A. Adaptive Image Contrast Enhancement Using Generalizations of Histogram Equalization. *IEEE Trans. Image Process.* **2000**, *9*, 889–896. [CrossRef] [PubMed]
23. Cao, G.; Tian, H.; Yu, L.; Huang, X.; Wang, Y. Acceleration of Histogram-Based Contrast Enhancement via Selective Downsampling. *IET Image Process.* **2018**, *12*, 447–452. [CrossRef]
24. Wu, H.T.; Huang, J.; Shi, Y.Q. A Reversible Data Hiding Method with Contrast Enhancement for Medical Images. *J. Vis. Commun. Image Represent.* **2015**, *31*, 146–153. [CrossRef]
25. Kim, S.; Lussi, R.; Qu, X.; Kim, H.J. Automatic Contrast Enhancement Using Reversible Data Hiding. In Proceedings of the 2015 IEEE International Workshop on Information Forensics and Security (WIFS), Rome, Italy, 16–19 November 2015; pp. 1–5. [CrossRef]
26. Gao, G.; Shi, Y.Q. Reversible Data Hiding Using Controlled Contrast Enhancement and Integer Wavelet Transform. *IEEE Signal Process. Lett.* **2015**, *22*, 2078–2082. [CrossRef]
27. Moorthy, A.K.; Bovik, A.C. Blind Image Quality Assessment: From Natural Scene Statistics to Perceptual Quality. *IEEE Trans. Image Process.* **2011**, *20*, 3350–3364. [CrossRef]
28. Chen, H.; Ni, J.; Hong, W.; Chen, T.S. Reversible Data Hiding with Contrast Enhancement Using Adaptive Histogram Shifting and Pixel Value Ordering. *Signal Process. Image Commun.* **2016**, *46*, 1–16. [CrossRef]
29. Wu, H.T.; Tang, S.; Huang, J.; Shi, Y.Q. A Novel Reversible Data Hiding Method with Image Contrast Enhancement. *Signal Process. Image Commun.* **2018**, *62*, 64–73. [CrossRef]

30. Ying, Q.; Qian, Z.; Zhang, X.; Ye, D. Reversible Data Hiding with Image Enhancement Using Histogram Shifting. *IEEE Access* **2019**, *7*, 46506–46521. [CrossRef]
31. Wu, H.T.; Mai, W.; Meng, S.; Cheung, Y.M.; Tang, S. Reversible Data Hiding with Image Contrast Enhancement Based on Two-Dimensional Histogram Modification. *IEEE Access* **2019**, *7*, 83332–83342. [CrossRef]
32. The USC-Sipi Image Database. Available online: http://sipi.usc.edu/database/ (accessed on 20 November 2021).
33. Kodak Lossless True Color Image Suite. Available online: http://www.r0k.us/graphics/kodak/ (accessed on 20 November 2021).
34. BOWS-2. Available online: http://bows2.gipsa-lab.inpg.fr (accessed on 10 January 2019).
35. Mittal, A.; Moorthy, A.K.; Bovik, A.C. No-Reference Image Quality Assessment in the Spatial Domain. *IEEE Trans. Image Process.* **2012**, *21*, 4695–4708. [CrossRef]
36. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef] [PubMed]